DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA 93943

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A COMPARISON OF TWO ACOUSTIC PARABOLIC EQUATION
TRANSMISSION LOSS MODELS FOR COMPATIBILITY
WITH THE WAVENUMBER TECHNIQUE IN THE
DETERMINATION OF SOURCE DEPTH

by

Joe Lane Blanchard II

March 1984

Thesis Advisor: A. B. Coppens

Approved for public release; distribution unlimited

T214376

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A Comparison of two Acoustic Parabolic Transmission Loss Models for Compatibility with the Wavenumber Technique in the Determination of Source Depth		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis March 1984
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Joe Lane Blanchard II		8. CONTRACT OR GRANT NUMBER(s) PE 63785N, NORDA
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS N6846283WR30089 of 29 APRIL 1983
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943		12. REPORT DATE March 1984
		13. NUMBER OF PAGES 151
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report)
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release, Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Parabolic Wave Equation, Underwater Acoustics, Implicit Finite-Difference, Split-Step Fast Fourier Transform, Wavenumber Technique, Source Depth Determination, Acoustic Propagation, ASW, Lloyd Mirror

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The Brock version of the Split-Step Fast Fourier Transform (SSFFT) and the Jeager version of the Implicit Finite Difference (IFD) acoustic parabolic equation models are compared with a Lloyd mirror interference pattern in the range domain. The SSFFT displays the inability to place the transmission loss nulls at the correct ranges. It is also unable to utilize bottom loss information correctly. The

IFD produced nulls at the correct ranges; however, it inserted an unaccepable amount of noise except when small (1 m) vertical grid steps were used and the pressure release bottom was placed at extended depths. In shallow water cases, the IFD is able to properly represent the pressure information. Each model is explored in the wavenumber domain by use of a "Wavenumber Technique" (WT) model with emphasis on source depth determination. The source depth may be determined by measuring the distance between the equally spaced nulls in the wavenumber representation. Neither acoustic model was able to provide accurate source depth information when the null spacings were compared to a known source-depth determination curve. Since the null spacings were not uniformly spaced, this was to be expected. Some specific problem areas in the models were identified by the use of the WT.

Approved for public release; distribution unlimited.

A Comparison of two Acoustic Parabolic Equation Transmission Loss Models for Compatibility with the Wavenumber Technique in the Determination of Source Depth

by

Joe L. Blanchard II
Lieutenant, United States Navy
B.S., University of North Carolina at Charlotte, 1974

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN METEOROLOGY AND OCEANOGRAPHY

from the

NAVAL POSTGRADUATE SCHOOL
March 1984

ABSTRACT

The Brock version of the Split-Step Fast Fourier Transform (SSFFT) and the Jeager version of the Implicit Finite Difference (IFD) acoustic parabolic equation models are compared with a Lloyd mirror interference pattern in the range domain. The SSFFT displays the inability to place the transmission loss nulls at the correct ranges. It is also unable to utilize bottom loss information correctly. The IFD produced nulls at the correct ranges; however, it inserted an unacceptable amount of noise except when small (1 m) vertical grid steps were used and the pressure release bottom was placed at extended depths. In shallow water cases, the IFD is able to properly represent the pressure information. Each model is explored in the wavenumber domain by use of a "Wavenumber Technique" (WT) model with emphasis on source depth determination. The source depth may be determined by measuring the distance between the equally spaced nulls in the wavenumber representation. Neither acoustic model was able to provide accurate source depth information when the null spacings were compared to a known source-depth determination curve. Since the null spacings were not uniformly spaced, this was to be expected. Some specific problem areas in the models were identified by the use of the WT.

TABLE OF CONTENTS

I. INTRODUCTION 11

II. WAVENUMBER TECHNIQUE (WT) 13

- A. BACKGROUND (LLOYD MIRROR) 13
- B. PHYSICAL PROCESS 16

III. ACOUSTIC MODEL ANALYSIS 22

- A. PARABOLIC EQUATION 22
- B. SSFFT 23
  - 1. Transmission Loss Comparison with Lloyd Mirror 23
  - 2. WT Analysis 28
- C. IFD 32
  - 1. Transmission Loss Comparison with Lloyd Mirror 32
  - 2. WT Analysis 39

IV. CONCLUSIONS 45

APPENDIX A: SSFFT SOURCE CODE (AT NPS) 47

APPENDIX B: WT SOURCE CODE 102

APPENDIX C: WT GRAPHICS SOURCE CODE 107

APPENDIX D: IFD SOURCE CODE (AT NPS) 112

LIST OF REFERENCES 147

BIBLIOGRAPHY 149

INITIAL DISTRIBUTION LIST 150

LIST OF FIGURES

2.1 Lloyd Mirror Geometry 13
2.2 Theoretical Lloyd Mirror Transmission Loss . . . 15
2.3 Quadrature Demodulation of a Signal 16
2.4 Example of WT Output at 50 Hz (From Lauer, 1979) 20
2.5 Source Depth Determination Curve 21
3.1 Test of SSFFT Model for Null Spacings 25
3.2 Test of SSFFT Model for Bottom Interaction . . . 27
3.3 SSFFT WT Plot with Source at 500 meters 29
3.4 SSFFT WT Plot with Receiver at 1000 meters . . . 30
3.5 SSFFT WT Plot with Source at 800 meters 31
3.6 IFD Environment 33
3.7 IFD Transmission Loss 34
3.8 IFD Detailed Transmission Loss 35
3.9 IFD Noise Free Transmission Loss 36
3.10 IFD: Shallow Water Transmission Loss 37
3.11 IFD: 500 meter Source, 15 meter Vertical Grid Step 40
3.12 IFD: 500 meter Source, 3.75 meter Vertical Grid Step 41
3.13 IFD: 500 meter Source, 0.8 meter Vertical Grid Step 42
3.14 IFD: Shallow Water WT, 1 meter Vertical Grid Step 44

Acronyms

AESD	Acoustic Environmental Support Detachment of the Office of Naval Research, now the Numerical Modeling Division (Code 320) of the Naval Ocean Research and Development Activity (NORDA).
ASTREX	Acoustic Storm Transfer and Response Experiment, conducted by the Naval Postgraduate School (NPS) in the northeastern Pacific during November and December of 1980.
NORDA	Naval Ocean Research and Development Activity at Bay St. Louis, Mississippi
NPS	Naval Postgraduate School at Monterey, California
NUSC	Naval Underwater Systems Center at New London, Connecticut

Symbols

d	Receiver Depth
f	Frequency
h	Source Depth
j	Square Root of -1
n	Index of Refraction (C_0/C)
m	Index in Calculations or Null Number in the Range Domain
p	Time Independent Factor of Complex Pressure
r	Distance of the Direct Path Wave
r	Distance to the Image (Reflected Path)
z	Source Depth
A	Amplitude
C	Sound Speed $(C(r,z))$
C_0	Reference Sound Speed (minimum sound speed in water mass profile)
F (K)	Pressure Field in the Wavenumber Space
FFT	Fast Fourier Transform
FFT^{-1}	Inverse Fast Fourier Transform
H	Hankel function
K	Wavenumber
K_0	Reference Wavenumber $(\omega/C\ )$
K_r	Horizontal Wavenumber

NPT	Number of Points in the Wavenumber Spectrum
P	Complex Pressure
R	Range between Source and Receiver
ΔR	Range Increment
$U(r,z)$	Envelope function
U_i	Envelope function (Imaginary part)
U_r	Envelope function (Real part)
Z_r	Receiver Depth
Z_s	Source Depth
β	Scaled Wavenumber
$\Delta\beta$	Scaled Wavenumber Increment
ω	Angular Frequency
∂	Partial Derivative Operator
π	3.1415...
∇	Laplacian Operator
$\sqrt{\ }$	Square Root Operator

ACKNOWLEDGEMENT

I would like to express my deepest appreciation to Dr. A. B. Coppens for his guidance in analytical processes and fundamentals of acousitcs. His understanding and encouragement provided the sound basis for this research effort. Dr. Coppens' knowledge of the subject matter combined with the ability to communicate this knowledge from many different perspectives is a unique talent recognized his students.

LCDR C. R. Dunlap, USN (Ret) was very supportive and encouraging during my entire thesis research effort. I appreciate his assistance with this project and his continuing work with the Environmental Acouctics Research Group.

I would like to thank the personnel of the W.R. Church Computer Center and especially Mr. D. Marr and Mr. J. Norton.

I. INTRODUCTION

Conventional methods used to represent underwater sound transmission have depicted transmission loss (dB) as a function of range in either tabular or graphical form. Since the interest was in determining quantities such as range of bottom bounce, convergence zones, and probability of detection, these methods proved very satisfactory. However when it became necessary to compare various underwater acoustic propagation loss models with each other or actual in-situ experiments, they provided the analyst with less than adequate insight into the causes for the observed errors in the various models outputs. Furthermore, interest has been generated in analyzing the received signal for information from the sound source which cannot easily be induced from transmission loss as a function of range.

One method, the Wavenumber Technique (WT), described by F. R. DiNapoli of NUSC [Ref. 1] and applied by Richard Lauer of NORDA [Ref. 2], provides more information for model comparisons and may have the capability of determining sound source depth and range from the receiver in certain cases. DiNapoli used an analysis of acoustic propagation in the wavenumber domain as an intermediate step in getting the transmission loss curve in the Fast Field Program (FFP). Lauer studied the wavenumber domain information from the FFP and concluded that source localization might be possible. B. B. Stamey [Ref. 3] investigated the potential for determining the source depth by using the Brock Split-Step Fast Fourier Transform (SSFFT) [Ref. 4] parabolic equation model with the "Wavenumber Technique" (WT) model. His preliminary investigation involved the use of isospeed and ASTREX sound speed profiles with a fully absorbing bottom, varied source/receiver combinations, and multiple frequencies.

The present investigation uses two acoustic parabolic equation models for comparison with theoretical Lloyd mirror depictions, and the effects of observed inconsistencies in predicted transmission loss curves on the calculated results from application of the WT model. An analysis of the sensitivity in determining source depth based on model output is attempted to ascertain possible use in naval operations.

To facilitate a straight-forward analysis of the results and evaluation of the WT technique for the cases studied, only a selected number of the various computer runs are included. Only those curves which best illustrate the effects revealed by this research are presented.

II. WAVENUMBER TECHNIQUE (WT)

A. BACKGROUND (LLOYD MIRROR)

The WT can be elucidated with the help of the classical Lloyd mirror effect [Ref. 5] which describes the interaction between direct path and surface reflected sound signals from the same continuous wave (CW) source. Figure 2.1 illustrates the geometry involved in the Lloyd mirror effect.

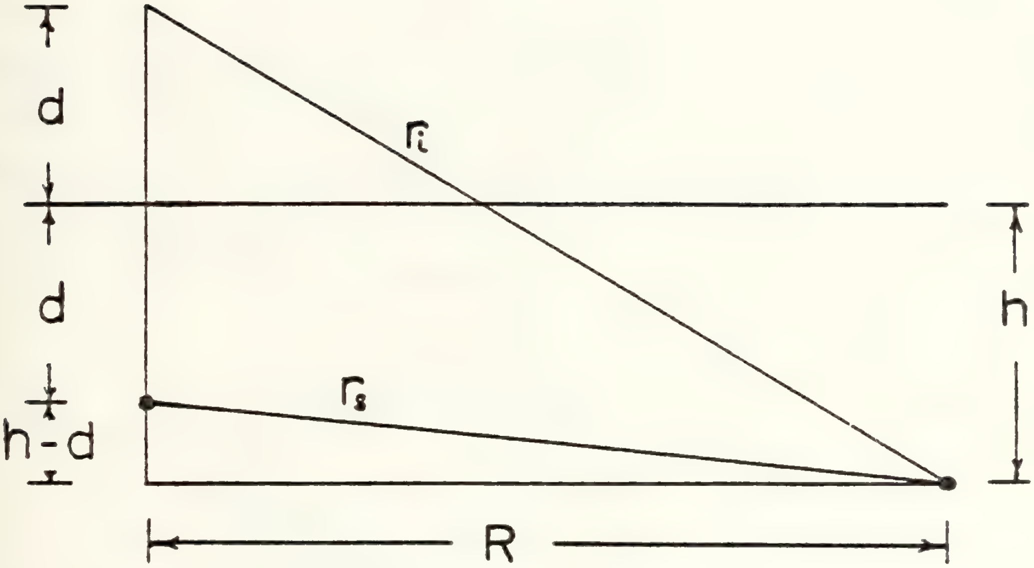

Figure 2.1 Lloyd Mirror Geometry.

The complex pressure (P) can be expressed by

$$P = A\left[\frac{1}{r_s}e^{-jkr_s} - \frac{1}{r_i}e^{-jkr_i}\right]\left(e^{+j\omega t}\right) \qquad \text{(eqn 2.1)}$$

when r_i and r_s are related by

$$r_s = \sqrt{R^2+(d-h)^2} \qquad r_i = \sqrt{R^2+(d+h)^2} \qquad \text{(eqn 2.2)}$$

with the assumption $r_i \sim r_s$ the pressure amplitude (p) can be approximated by

$$R_m = \frac{2fhd}{mC}\ , m=1,2,3,\ldots \qquad \text{(eqn 2.3)}$$

The interaction between the direct path and surface reflected waves produces constructive and destructive interference which is manifested as peaks and nulls for the in phase and out of phase conditions respectively. Figure 2.2 was produced by using equation 2.3. The null locations can be obtained by the relationship.

$$p = \frac{2A}{R}\sin\left(\frac{khd}{R}\right) \qquad \text{(eqn 2.4)}$$

When equation 2.3 is expressed in the horizontal wavenumber domain (k_r), the details of the resultant functional dependency on horizontal wavenumber yield useful information.

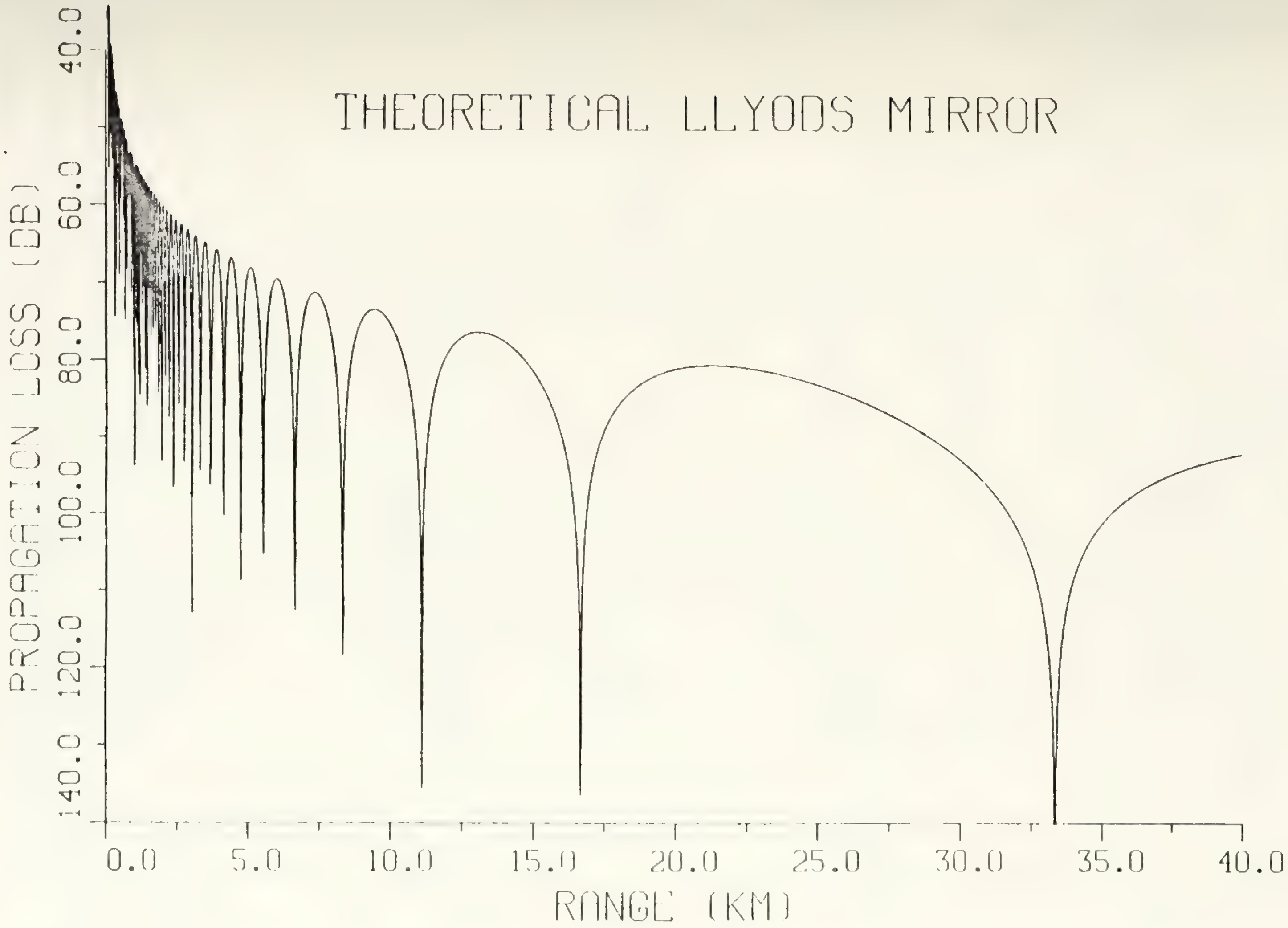

Figure 2.2 Theoretical Lloyd Mirror Transmission Loss.

B. PHYSICAL PROCESS

The WT is a process by which the complex pressure wave (with range dependent amplitude and phase) is transformed into the spectral density as a function of the horizontal wavenumber. The use of the WT in the operational environment would require a quadrature demodulation of the source signal to attain the complex pressure. The quadrature demodulation process can best be defined by the illustration in figure 2.3 [Ref. 6].

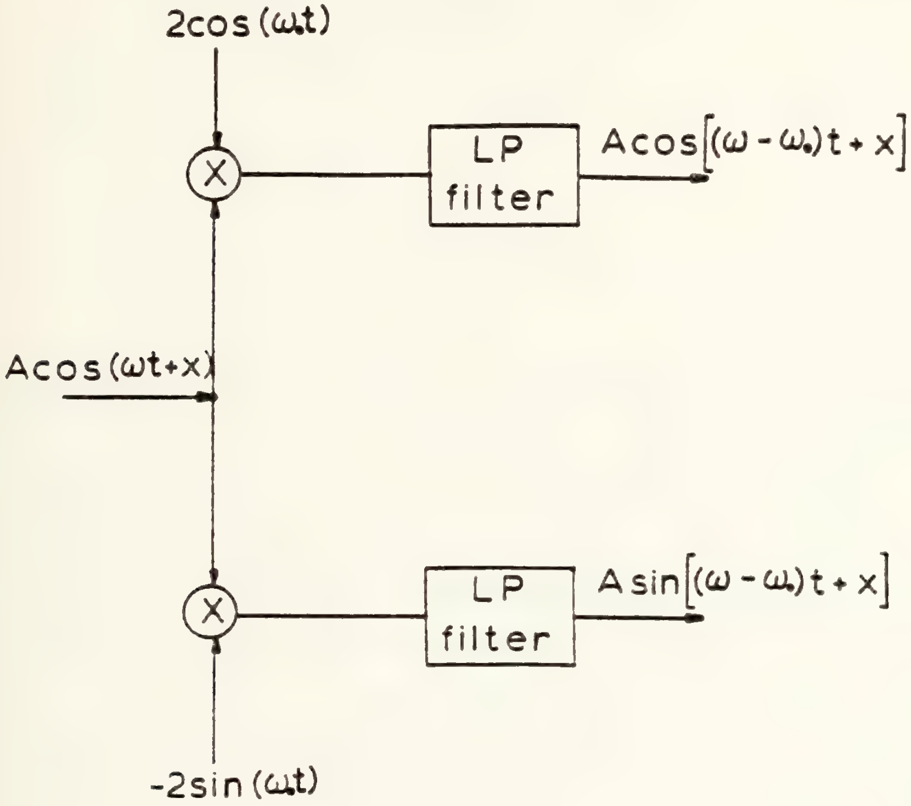

Figure 2.3 Quadrature Demodulation of a Signal.

When acoustic models are used to simulate the environment, the requirement to perform a quadrature demodulation is eliminated since the complex pressure is directly

accessible. In actuality, acoustic parabolic equation models provide the complex envelope function (U) which must be multiplied by the Hankel function in order to obtain the complex pressure. The acoustic wave is written in the form

$$P = U H_0^{(1)}(k\ r)\, e^{-j\omega t} \qquad \text{(eqn 2.5)}$$

where the reference wavenumber is defined by

$$k_o = \frac{\omega}{C_o} \qquad \text{(eqn 2.6)}$$

and

$$U = U(r, z) \qquad \text{(eqn 2.7)}$$

is the solution to the appropriate parabolic equation. The complex pressure is corrected for volume attenuation and Fourier transformed to attain the spectral density. A plot displaying the results graphically with the horizontal wavenumber on the x-axis and the normalized spectral density on the y-axis is constructed and the spacings between the nulls measured. Figure 2.4 [from Ref. 2] is an example of the WT for a source and receiver in the same type of water mass.

Since the Lloyd mirror field in the wavenumber domain is given by [Ref. 2]

$$F(k) = \frac{\sin(\beta Z_s)}{\beta}\, e^{j(\beta Z_r)} \qquad \text{(eqn 2.8)}$$

an alternative formulation [Ref. 2], which produces evenly spaced nulls, is obtained by converting the horizontal wavenumber to beta

$$\beta = \sqrt{k_o^2 - k_r^2} \qquad \text{(eqn 2.9)}$$

By using beta instead of the horizontal wavenumber, the distance, delta beta, between any two nulls can be used to ascertain the source depth by

$$\Delta\beta = \frac{\pi}{z_s} \qquad \text{(eqn 2.10)}$$

as illustrated in figure 2.5.

The application for implying the method of images, in isospeed cases, to describe the effects of a perfectly flat pressure release boundary can be justified by inspection. Therefore the parabolic equation can be used to produce the Lloyd's mirror effect and surface decoupling is not an issue, cf. Chapter 4 of Brekhovskikh [Ref. 7].

When the WT is implemented, care must be exercised to ensure that the complex pressure contains an adequate number of points to describe the shortest periodicity in the pressure field. This will preclude the possibility of aliasing in the wavenumber domain. The real and imaginary elements of the pressure signal must be modified so that the signal begins and ends with zero values. This modification is necessary because the pressure signal must represent a repetitive oscillation for the transform. The complex pressure array is zero-filled beyond the data in order to generate an array which has the length of a power of 2 for the Fast Fourier Transform (FFT). The horizontal wavenumber increment is generated within the computer code by

$$k_{rm} = k_o + \frac{2\pi}{\Delta r}\left(\frac{m}{NPT} - 1\right) \qquad \text{(eqn 2.11)}$$

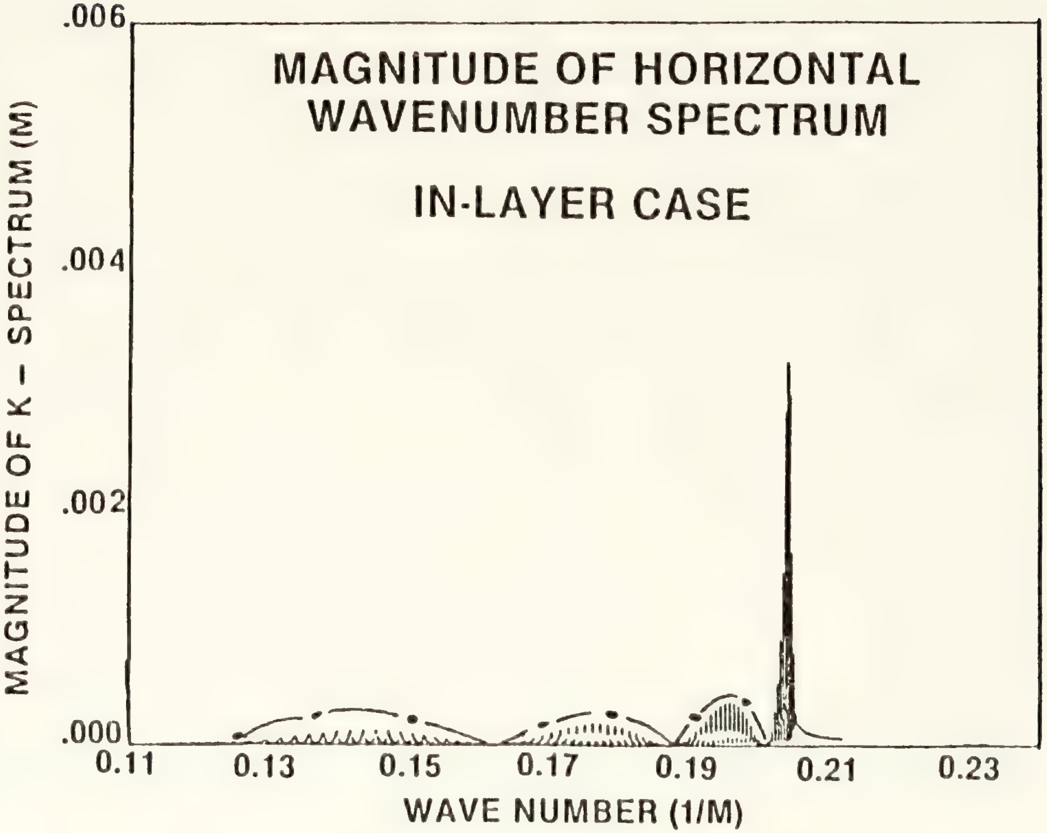

Figure 2.4 Example of WT Output at 50 Hz (From Lauer, 1979).

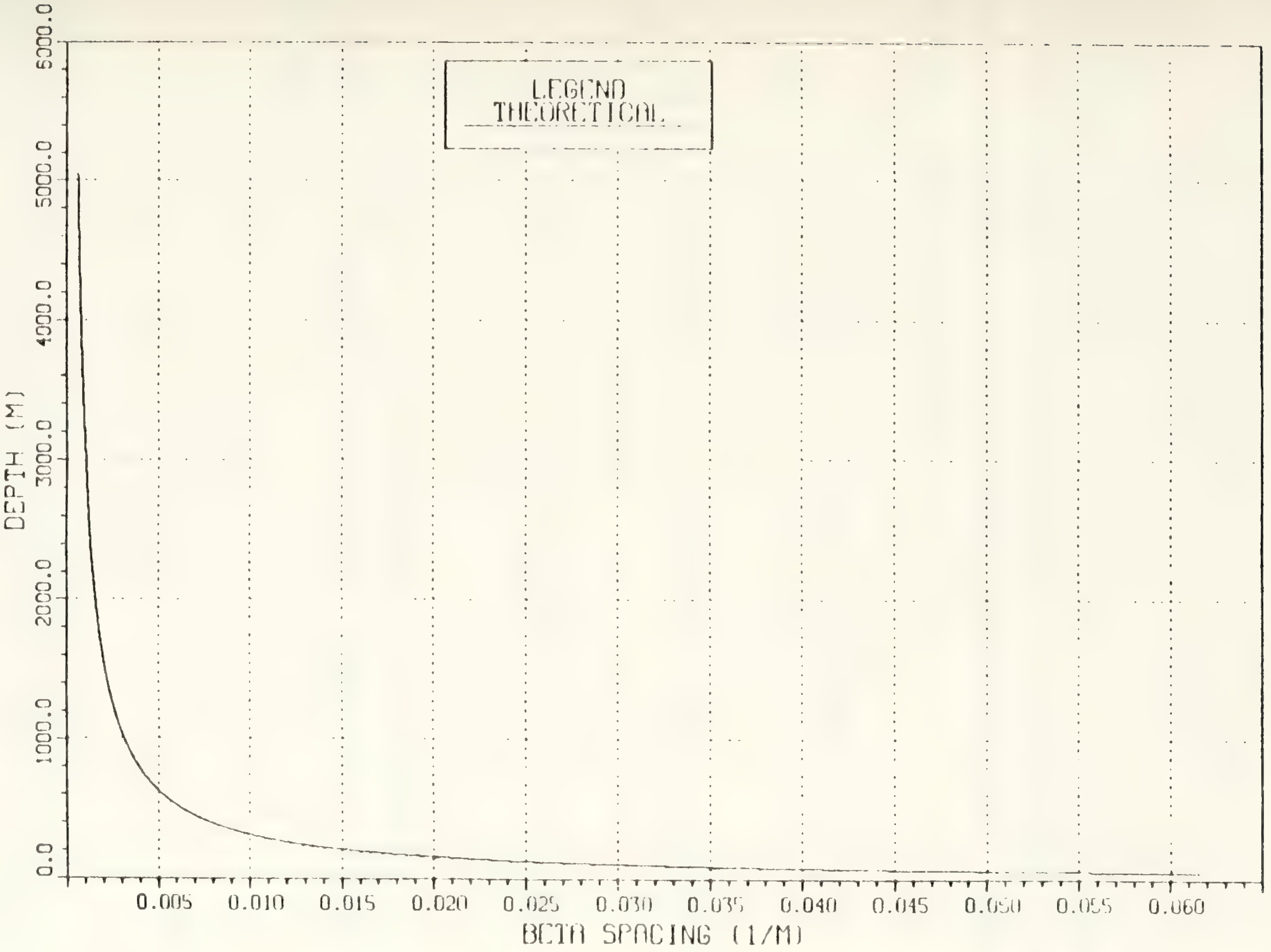

Figure 2.5 Source Depth Determination Curve.

III. ACOUSTIC MODEL ANALYSIS

A. PARABOLIC EQUATION

The elliptical wave equation can be approximated by a parabolic equation (PE) when it is assumed that the envelope function varies slowly with range. A detailed mathematical description of the PE from an acoustic point of view is presented by DeSanto [Ref. 8 and 9] and a simpler but less general description can be found in Coppens [Ref. 10]. The parabolic equation has the form

$$\frac{\partial^2 U}{\partial z^2} + 2jk_o\left(\frac{\partial U}{\partial r}\right) + (k^2 - k_o^2)U = 0 \qquad \text{(eqn 3.1)}$$

where it is assumed that the pressure has the form of equation 3.2,

$$p = U(r,z)\,S(r) \qquad \text{(eqn 3.2)}$$

S(r) represents the primary radial dependence of the field in terms of an outward propagating cylindrical wave of the form [Ref. 4]

$$S(r) = H_o^{(1)}(k_o r) \qquad \text{(eqn 3.3)}$$

If it is assumed that the range of interest is many wavelengths from the source; then the asymptotic form of the Hankel function (equation 3.4) can be used [Ref. 11].

$$H_0^{(1)} \approx \sqrt{\frac{2}{\pi k_0 r}}\; e^{j(k_0 r - \pi/4)}\ , \quad k_0 r >> 1 \qquad \text{(eqn 3.4)}$$

Two PE models, the Brock version of the Split-Step Fast Fourier Transform (SSFFT) and the Jeager version of the Implicit Finite Difference (IFD) were investigated for compatibility with the WT in determination of source depth.

B. SSFFT

1. Transmission Loss Comparison with Lloyd Mirror

To analyze sound waves in the wavenumber domain, we needed a model to provide the pressure as a function of range. Initially the SSFFT model [Ref. 4] was selected. This model generates successive values for U as a function of range with the help of the algorithm

$$U(r+\Delta r, z) = e^{j\Delta r k_0 (n^2-1)/2} \qquad \text{(eqn 3.5)}$$
$$FFT^{-1}\left\{ e^{j\Delta r k^2/2k_0}\, FFT(U(r,z)) \right\}$$

This model was a natural choice since it was the model used in the preliminary study dealing with the determination of the source depth by the WT [Ref. 3]. However, it soon became apparent that this model's inherent weaknesses would require the consideration of another model if ocean bottom interactions were to be studied. These weaknesses, and their impact upon the WT, will be discussed shortly.

The SSFFT is a range dependent acoustic wave model which, for this analysis, will be operated in a range independent manner. In other words, only one sound speed profile will be used, the water mass will be assumed homogeneous, and the bottom flat. To facilitate the study of the SSFFT model, the source code of the model was modified to generate the variables required for follow-on programs to

transform the pressure information and then display the results graphically. A copy of the source code listing for each program is provided in appendices A, B, and C. These programs were then linked by Job Control Language (JCL) so that the WT graphics were an automatically executed.

A test run of the SSFFT was made for 100 hz, 500 m source and receiver depths, 0.019 km range step, fully absorbing bottom, 2000 m water depth, and an isospeed profile at 1500 m/sec. These values were chosen so that a comparison could be made with the theoretical Lloyd mirror transmission loss (figure 2.2). The output was displayed as a transmission loss curve (figure 3.1). From equation 2.4, the location of null number 1 (R1) should be 33.33 km and null number 2 (R2) should be at 16.67 km. The SSFFT placed R1 at approximately 38.3 km and R2 at 17.1 km. Since the model kept the frequency, source depth, and receiver depth constant, then this equates to a reference sound speed of 1305 and 1462 m/sec respectively. It is apparent that as the range decreases the error is reduced which indicates the possibility of a problem in the range step section of the model which leads to the introduction of a systematic and increasing deviation of the calculated sound field from that predicted by the classical Lloyd's mirror interference pattern. The "washing out" of the interference pattern for ranges less than about 2 km is a result of the particular approximation of a point source used to initiate the program, which is to be expected. Rather regular fluctuations develop beyond about 30 km; from the geometry of the case, it is plausible that this arises from interference with bottom reflected signals.

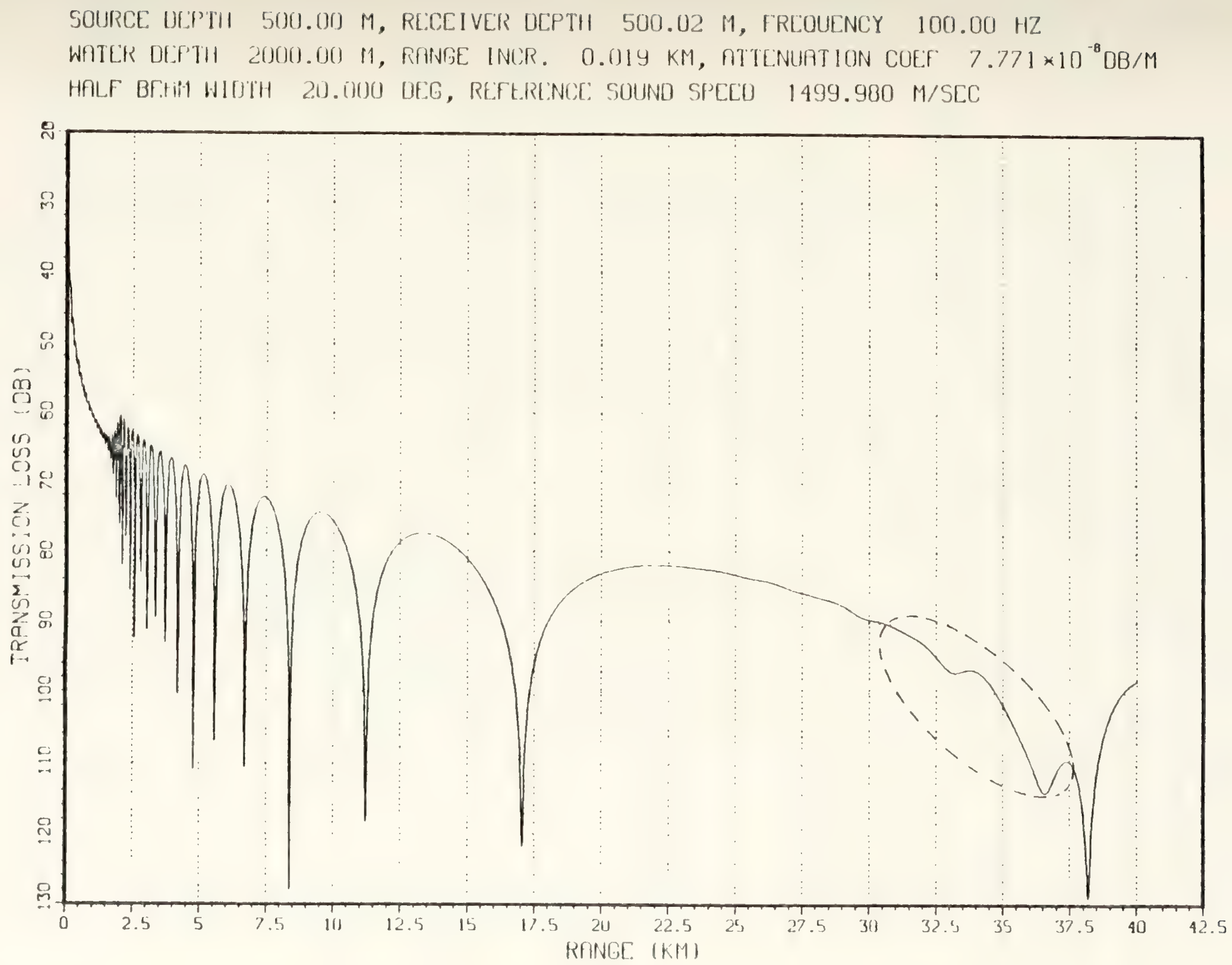

Figure 3.1 Test of SSFFT Model for Null Spacings.

Another test was made using the same inputs as above except that the source and receiver were moved to a depth of 1500 m (figure 3.2). Since the half beamwidth is 20 degrees and the bottom is fully absorbing, the shortest range at which a surface reflection could occur is approximately 8.2 km; beyond this range, surface reflections should be spaced in accordance with equation 2.4. The nulls beyond 8.2 km do occur at the correct ranges; however, there are slow modulations of the overall transmission loss signal. These slow modulations are another indication that the bottom is not fully absorbing as had been specified as input to the program.

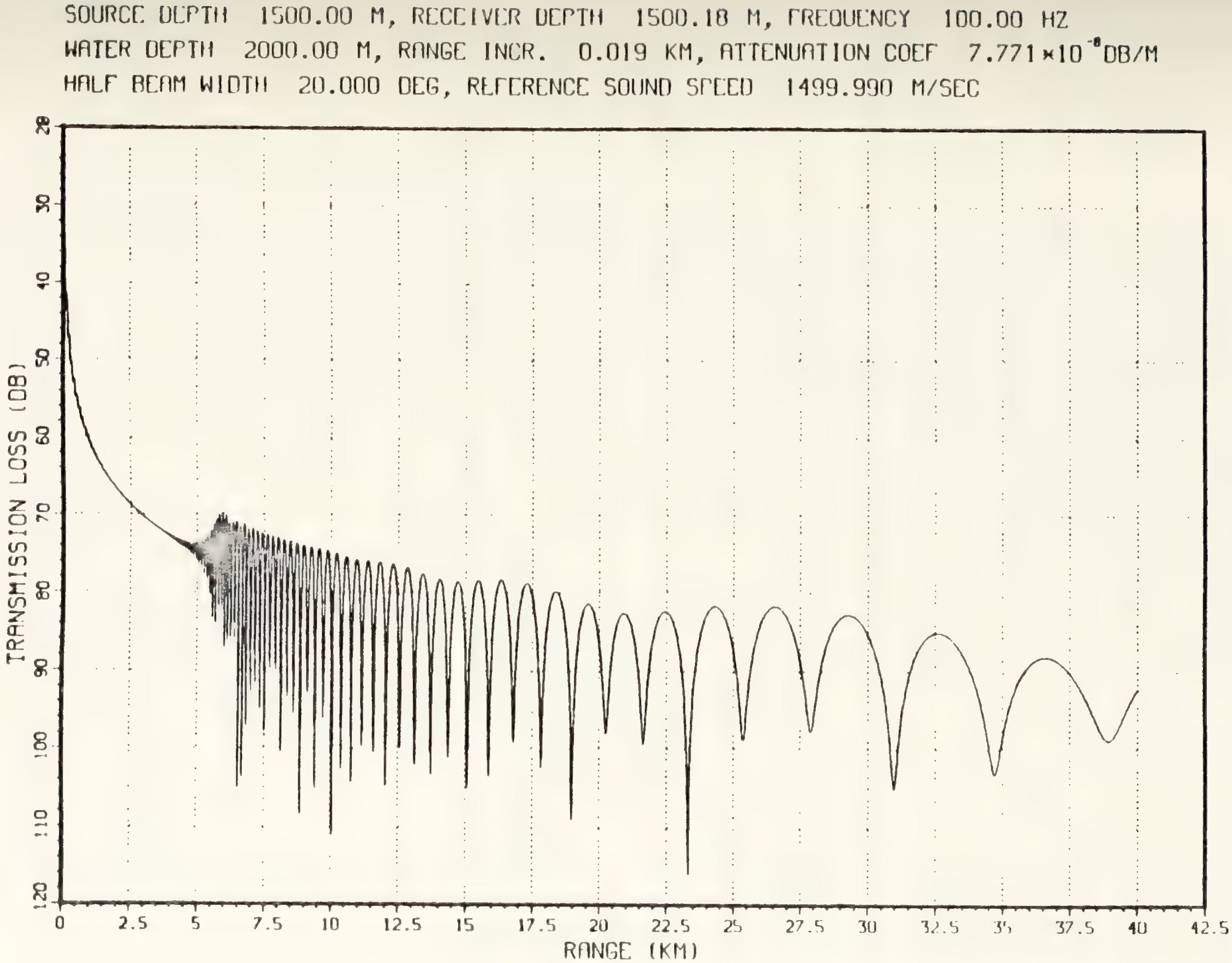

Figure 3.2 Test of SSFFT Model for Bottom Interaction.

2. WT Analysis

The WT output from the SSFFT model is in English units, which is a problem when comparing different models. However, for this investigation the problem was minimal. As expressed by equation 2.6, the reference wavenumber for 100 hz with a reference sound speed of 4921.26 ft/sec (1500 m/sec) is 0.1277 1/ft. After correction of the reference wavenumber for a range increment of 62.3 ft and the 2048 points used to represent the wavenumber spectrum, the maximum beta value should be 0.1248 1/ft which agrees reasonably well with the maximum beta values in figures 3.3, 3.4, and 3.5. Contrary to theory the nulls are not equally spaced and the expected delta beta value of 0.00192 1/ft (equation 2.10) is not represented by any of the null spacings. Stamey [Ref. 3], observed the same non-uniform spacing of beta nulls. The reason why the source depth is not represented by any of the null spacings may be traced to the improper null placement in the range domain. The lack of equal spacing between the nulls must be attributed to something else, possibly the pressure wave was improperly represented as a continuous signal or numerical errors exist within the source code. Whatever the reason for this difference, it must be resolved before the SSFFT can be used with the WT to determine the source depth.

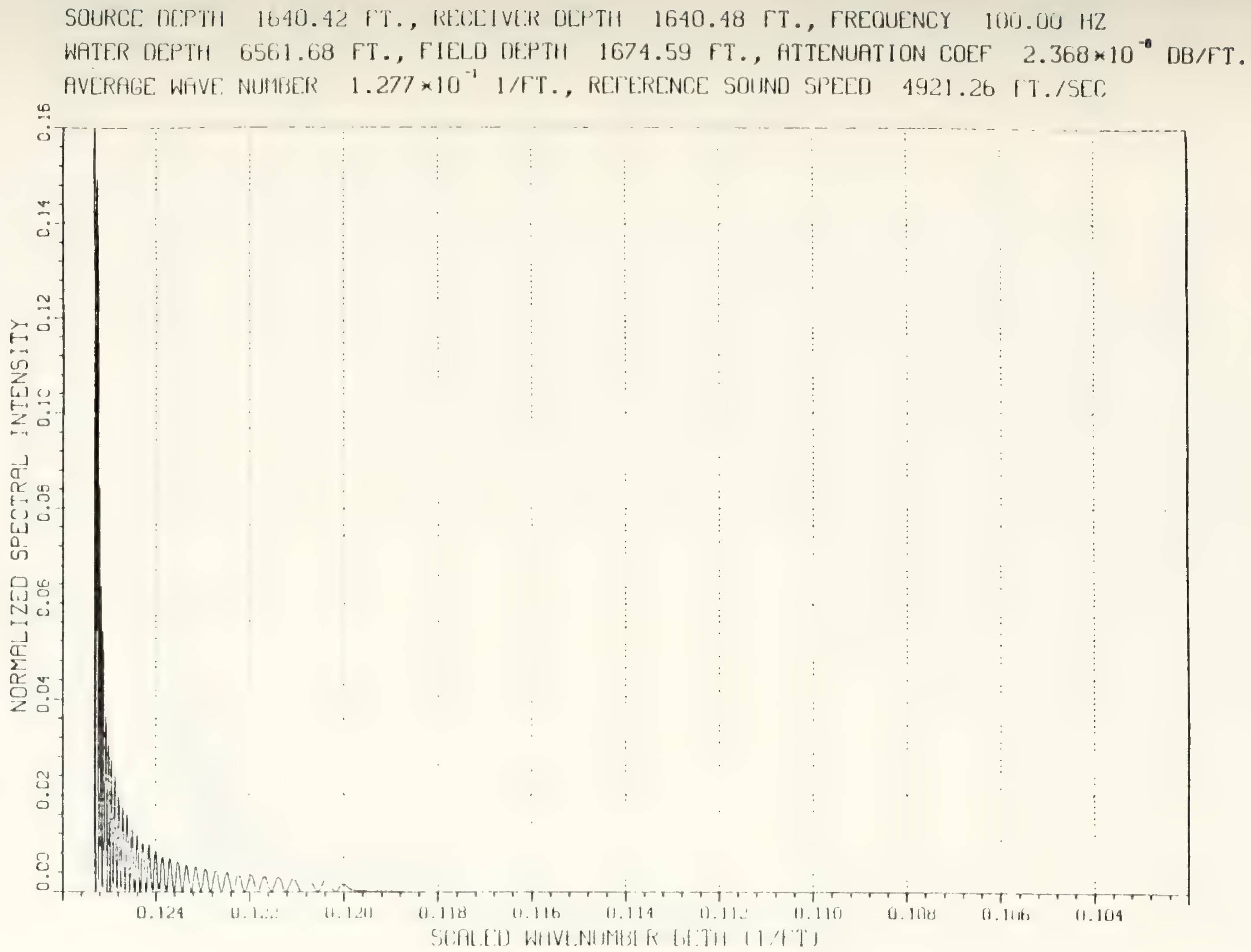

Figure 3.3 SSFFT WT Plot with Source at 500 meters.

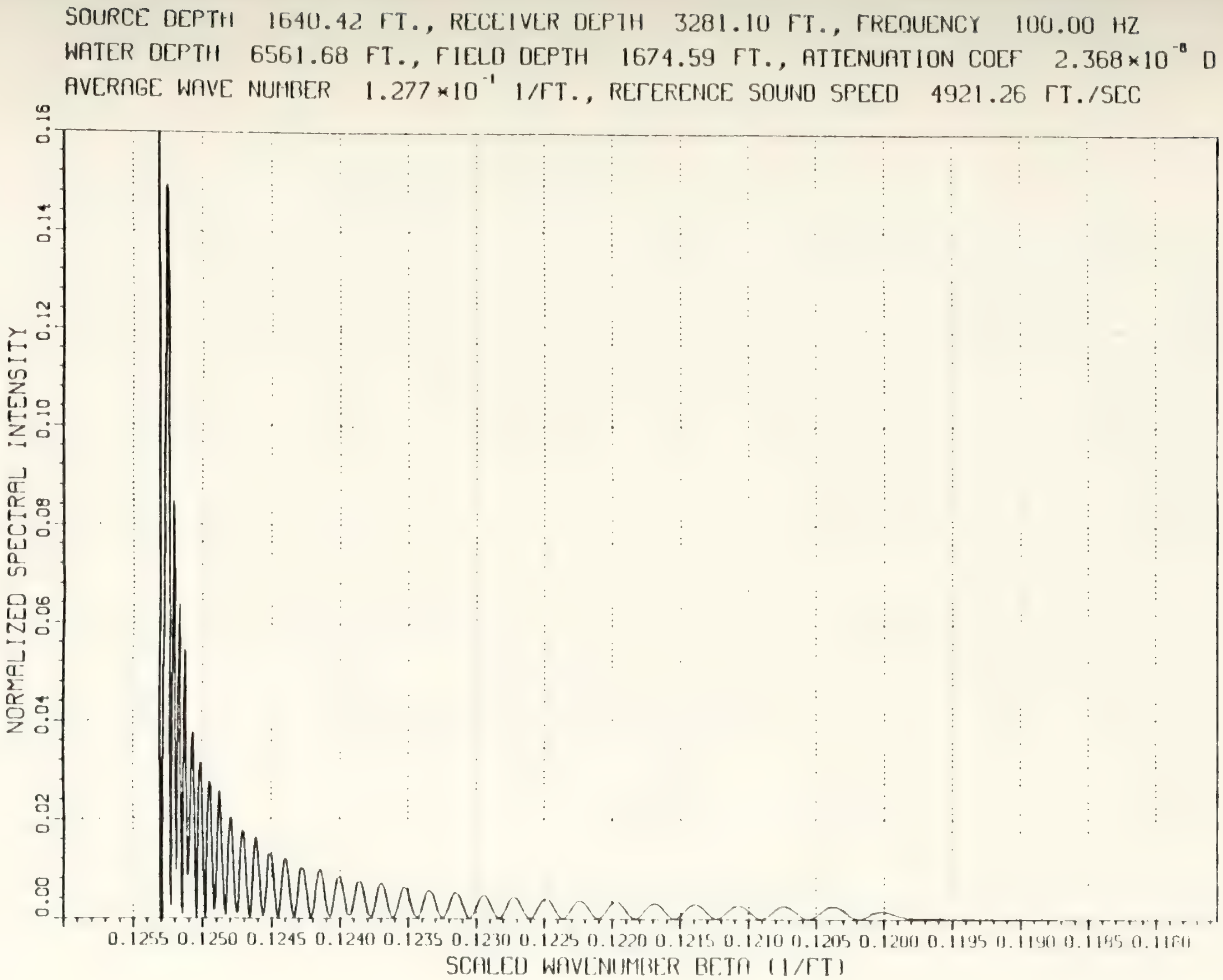

Figure 3.4 SSFFT WT Plot with Receiver at 1000 meters.

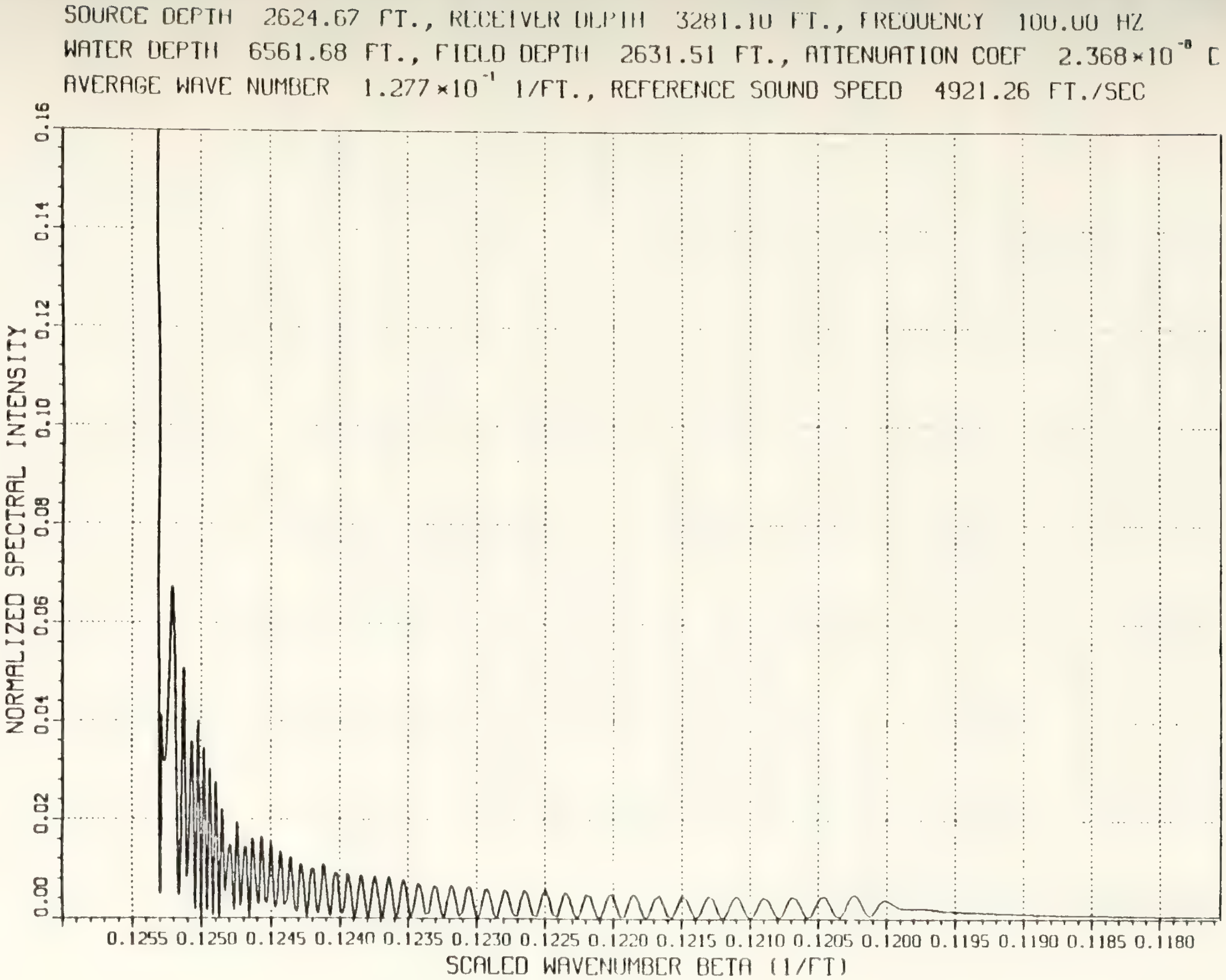

Figure 3.5 SSFFT WT Plot with Source at 800 meters.

IFD

1. Transmission Loss Comparison with Lloyd Mirror

The IFD model [Ref. 12] which was installed and ested in September 1983 at NPS [Ref. 13] was the next model hosen as a sound source for the WT. This decision was made ecause the IFD was capable of properly handling the ocean ottom interactions and was the only other acoustic PE model in residence at NPS. A copy of the source code listing is provided in appendix D.

The IFD also presented problems which will be discussed further. The same initial test run used with the SSFFT was used with the IFD. Since the IFD requires more bottom information, additional testing was performed in order to obtain a fully absorbing bottom. Figure 3.6 illustrates the environment as seen by the IFD model.

The densities in the water mass, sediment, and artificial attenuation layer were set to a constant of 1.0 and the attenuation in the sediment was adjusted until the elimination of bottom interference was observed. Figure 3.7 is an example of the IFD transmission loss output for a 500 m source/receiver depth, a water depth of 2000 m, an upper surface of the artificial attenuation layer at 3000 m, and a lower pressure release surface at 4000 m. The bottom attenuation used to produce figure 3.7 was 0.0016 dB/wavelength.

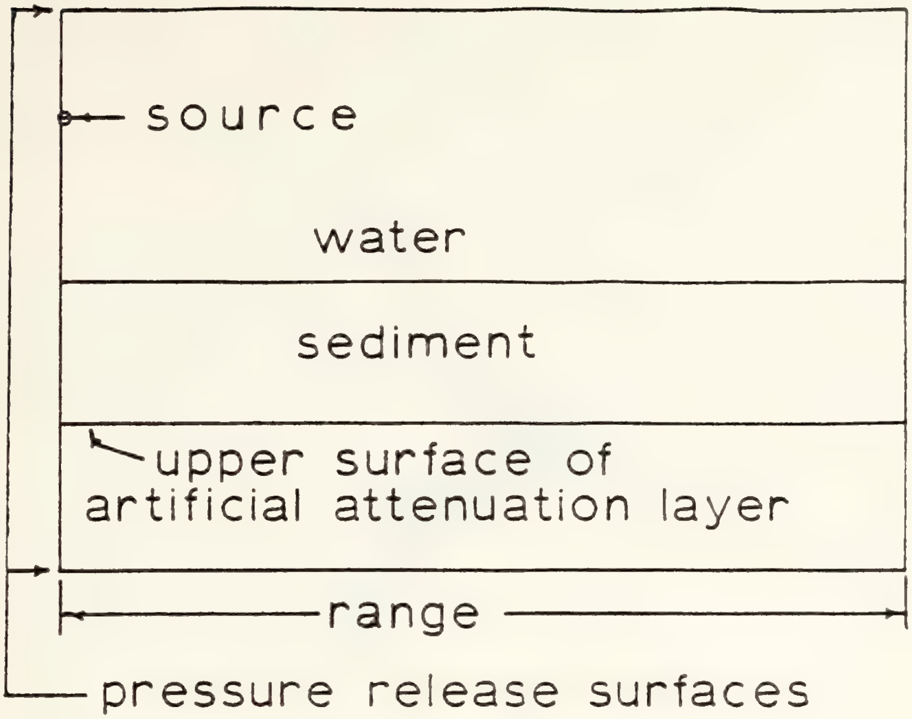

Figure 3.6 IFD Environment.

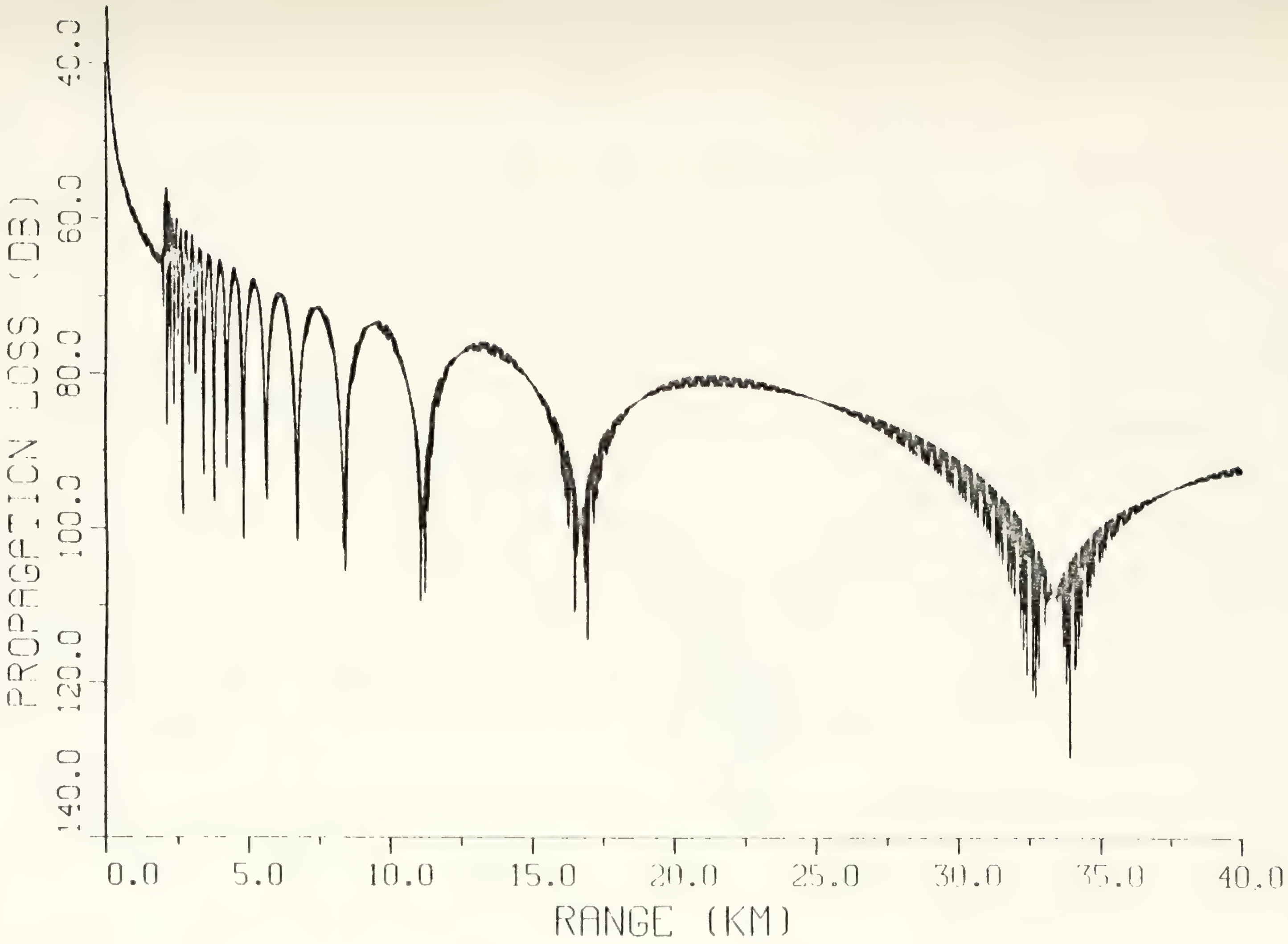

Figure 3.7 IFD Transmission Loss.

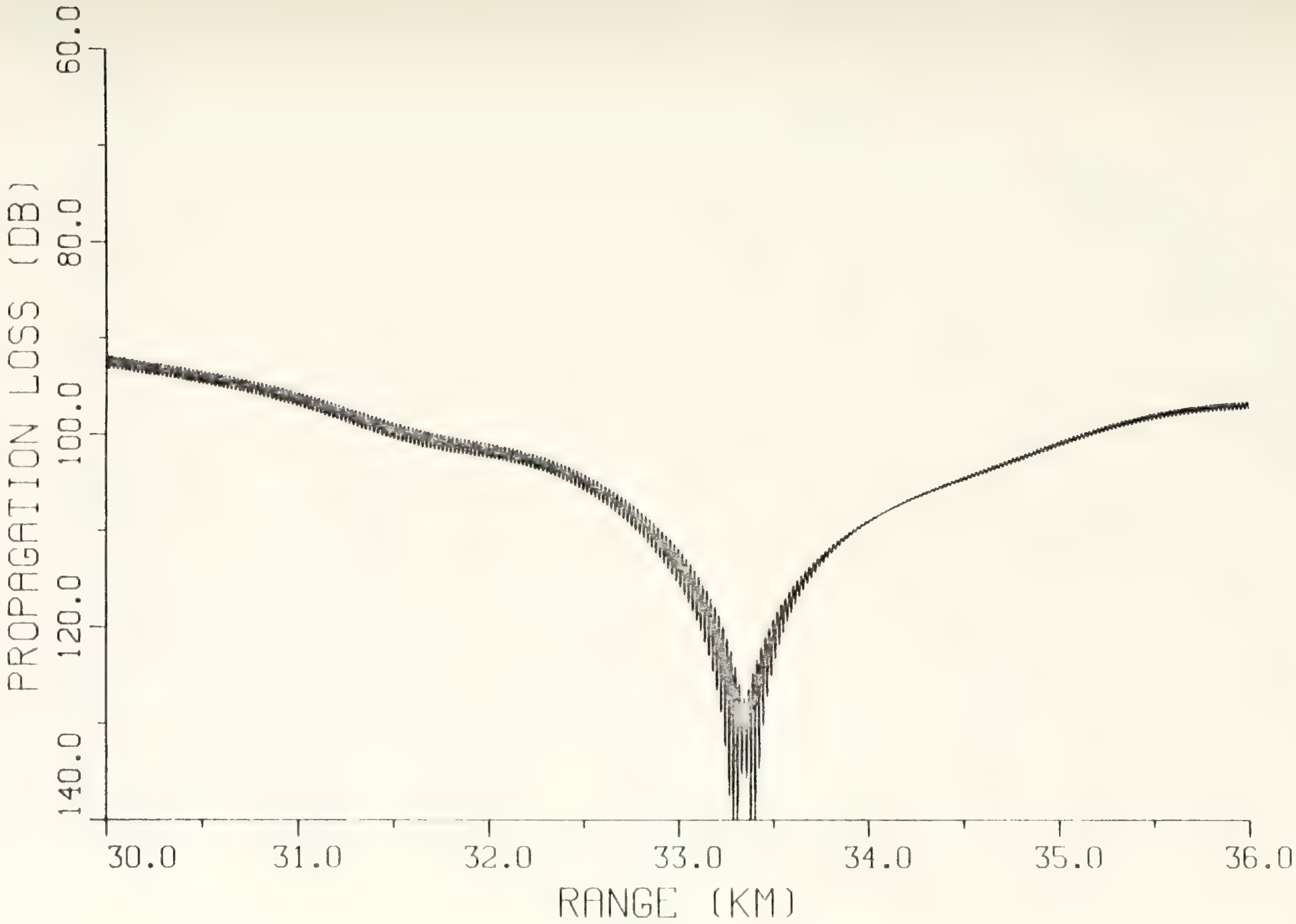

Figure 3.8 IFD Detailed Transmission Loss.

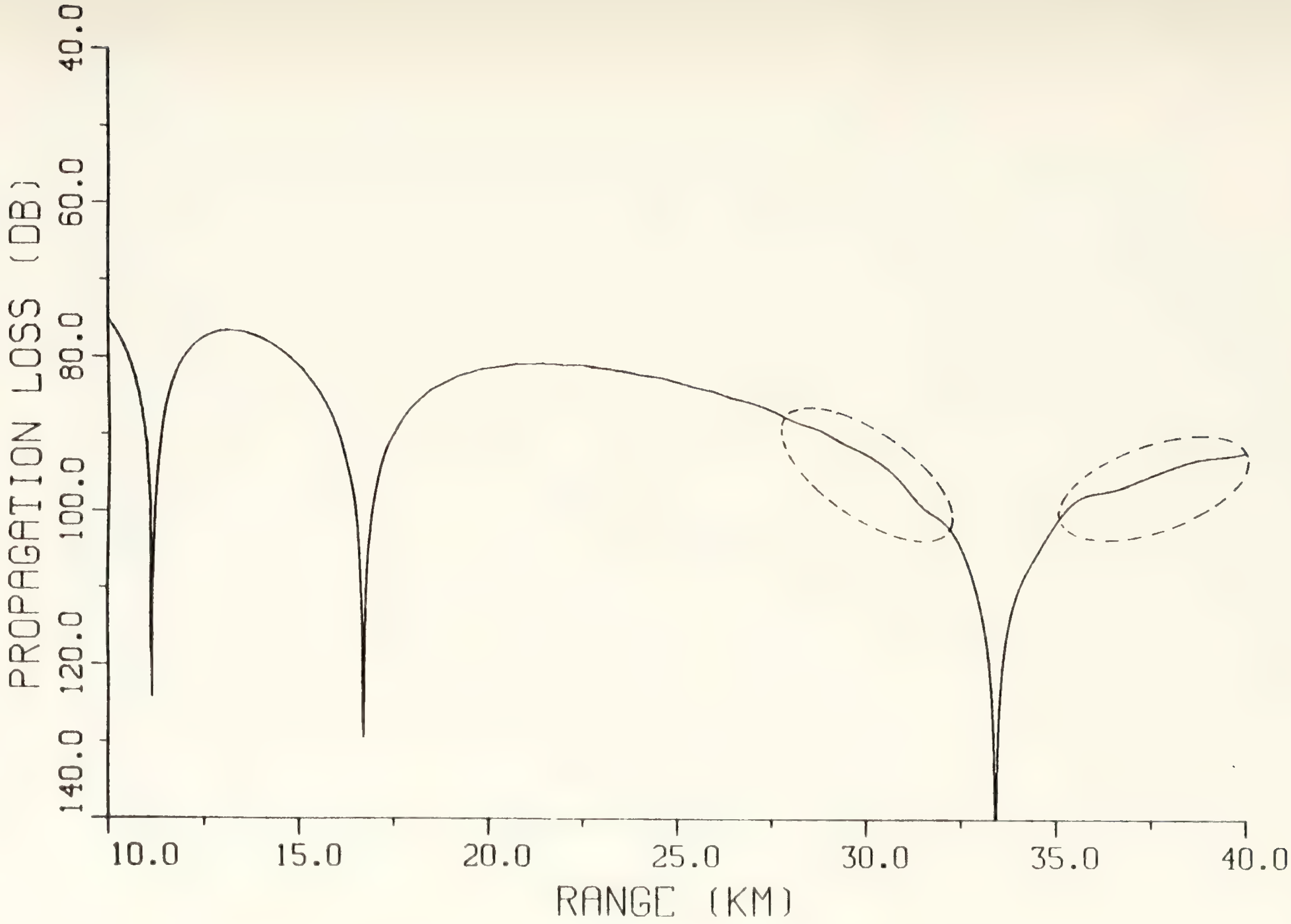

Figure 3.9 IFD Noise Free Transmission Loss.

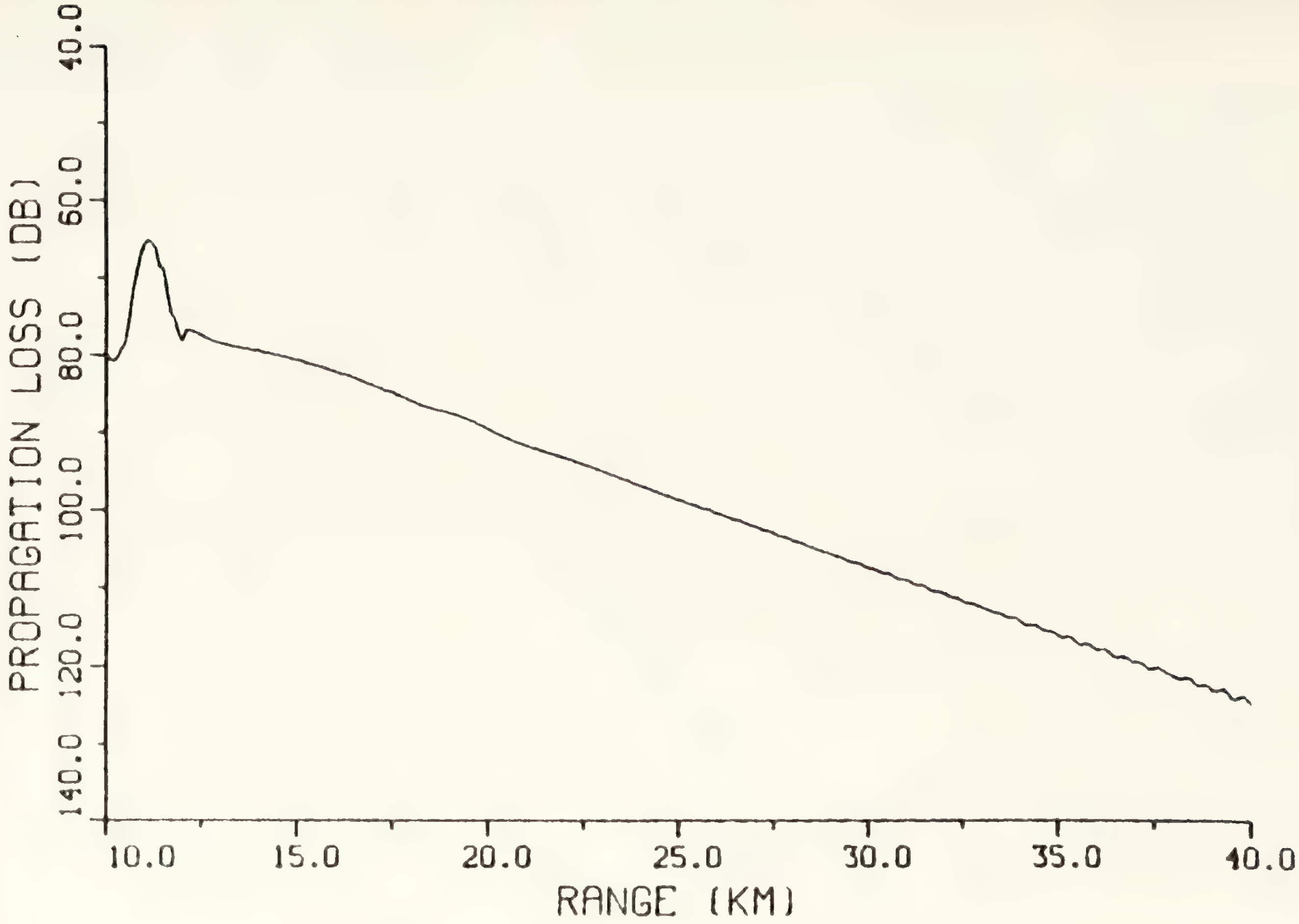

Figure 3.10 IFD: Shallow Water Transmission Loss.

The location of the nulls in figure 3.7 coincide with those predicted by equation 2.4 if the high frequency noise is filtered out by either a running average process or low pass digital filter. A careful examination of the transmission loss curve (figure 3.8) reveals the real problem, which is the long wavelength noise near the wavelength of interest. The long wavelength noise will cause erroneous spikes on the WT plots after being Fourier transformed. Continued analysis revealed that the noise could be eliminated if the thicknesses of the artificial attenuation layer and the bottom pressure release surface are chosen so that they are at least twice the water depth. The choice of a small vertical grid step will reduce the thickness required for the pressure release surface, but this version of the IFD program is limited to 5000 vertical grid points due to software restrictions. Figure 3.9 is a transmission loss curve obtained from a 500 m source/receiver depth, a 2000 m water depth, a 4000 m upper level artificial attenuation layer, and an 8000 m bottom pressure release surface. Comparison of figures 3.1 and 3.9 with the classical Lloyd's mirror transmission loss (figure 2.2) shows that the shape of the curve beyond 30 km is greatly improved with the IFD. The number of vertical grid steps was 5000. When the IFD is used in a shallow water situation and at 25 hz, the noise does not appear to be present. Figure 3.10 is a plot of the transmission loss for a signal at 25 hz, with the source/receiver depth at 50 m, the bottom sloping up from 350 to 50 m, and an artificial attenuation layer and bottom pressure release surface located at 750 and 1000 m, respectively. The noise which was so evident in figure 3.7 is absent from the shallow water case. Producing the noise free curves requires excessive computer time except for shallow water cases.

2. WT Analysis

To reduce the amount of the pressure information and yet preserve the details, the IFD tests were conducted between 10 and 40 km in range. The reference wavenumber for the test input with the IFD was 0.4189 1/m which, after correcting for a range increment of 7.5 m with 4096 points, produced a beta maximum of 0.4188 1/m. Figures 3.11, 3.12, and 3.13 indicate agreement with the beta maximum and a delta beta of 0.00628 1/m can be found among the null spacings. When the vertical grid step is reduced in size, the output signal contains more information of finer detail. The finer detailed information will, if noise is present, produce more clutter on the WT depiction (figure 3.13). In numerous WT depictions a peak was observed at the minimum and maximum wavenumbers. Stamey believed that the right and left intensity maxima corresponded to beam elevation angles of 0 and 30 degrees, respectively. He further stated that the left maxima also may be related to algorithmic inaccuracies [Ref. 3]. The "U-shaped phenomenon" observed during Stamey's investigation of the SSFFT, is present with the IFD and the nulls are not equally spaced.

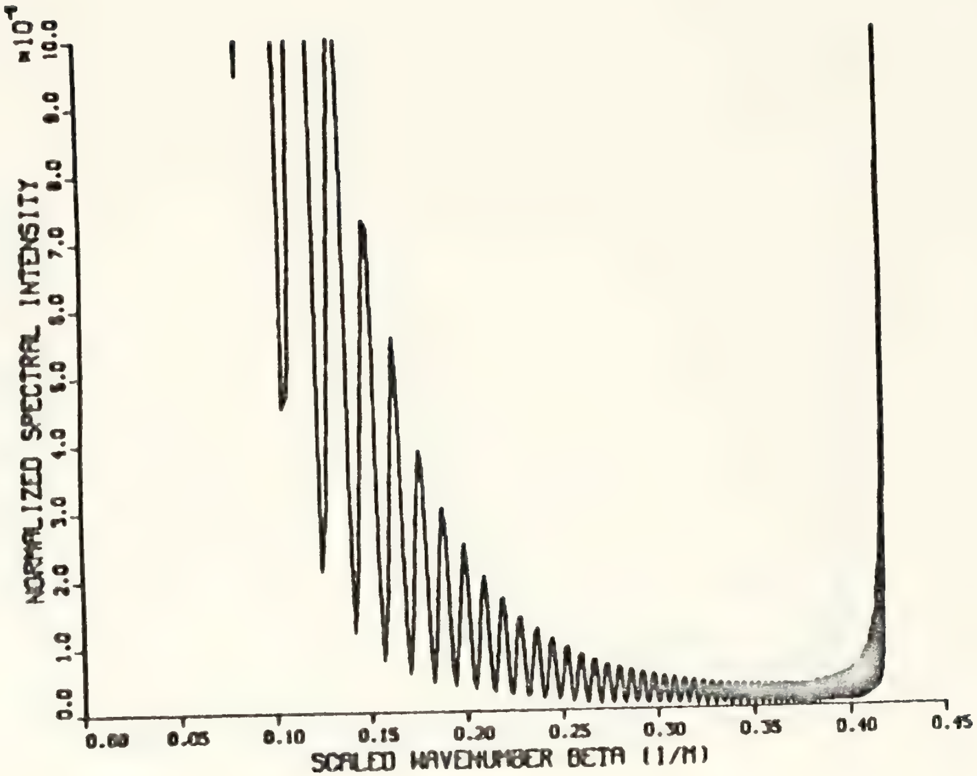

Figure 3.11 IFD: 500 meter Source, 15 meter Vertical Grid Step.

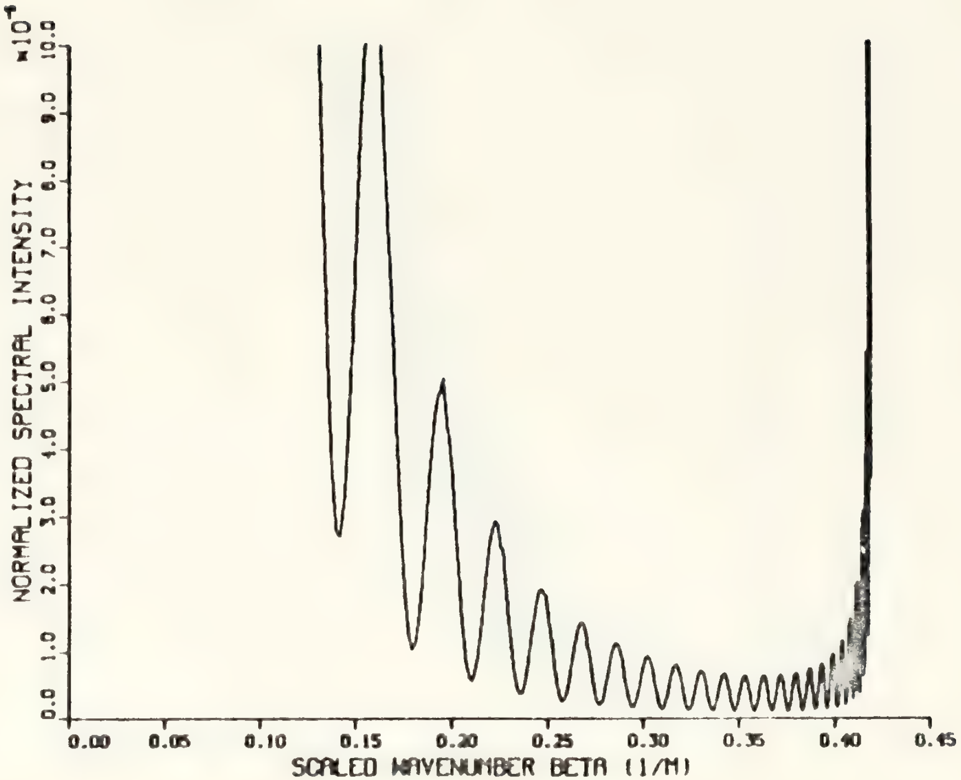

Figure 3.12 IFD: 500 meter Source, 3.75 meter Vertical Grid Step.

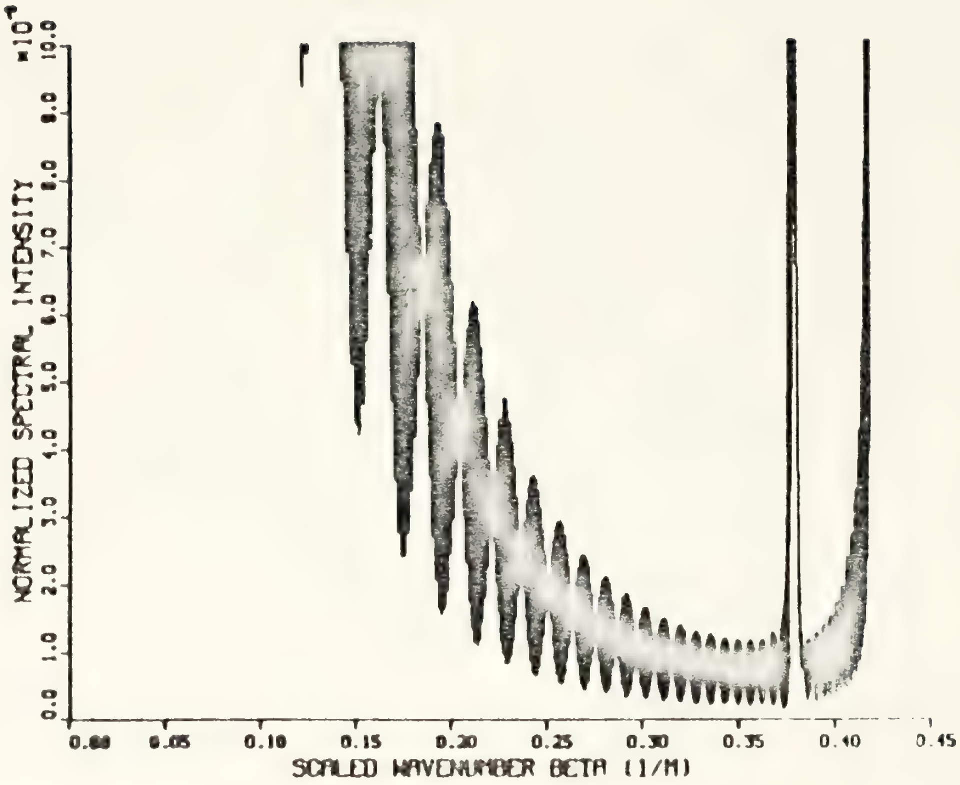

Figure 3.13   IFD: 500 meter Source, 0.8 meter Vertical Grid Step.

The reason for the "U-shaped phenomenon" appears to be related to noise in the envelope function field which is further amplified when the Hankel function is inserted to obtain the complex pressure. Using a low pass filter and increasing the vertical grid step size appear to reduce the problem. In figure 3.8, the nulls occur at 33.30 and 33.39 km when a single null should exist at 33.33 km. This condition or noise produces the most devastating effects by masking the desired density spectrum. A narrow band pass filter would probably solve this problem in the short term but eventually the IFD will require further study to eliminate the noise. When the shallow case is used in the WT (figure 3.14), the "U-shaped phenomenon" is absent and the null locations are more distinct. However, in the shallow water case, the bottom interaction produces high frequency interference but the null spacings occur near the expected value for the source depth (0.0628 1/m).

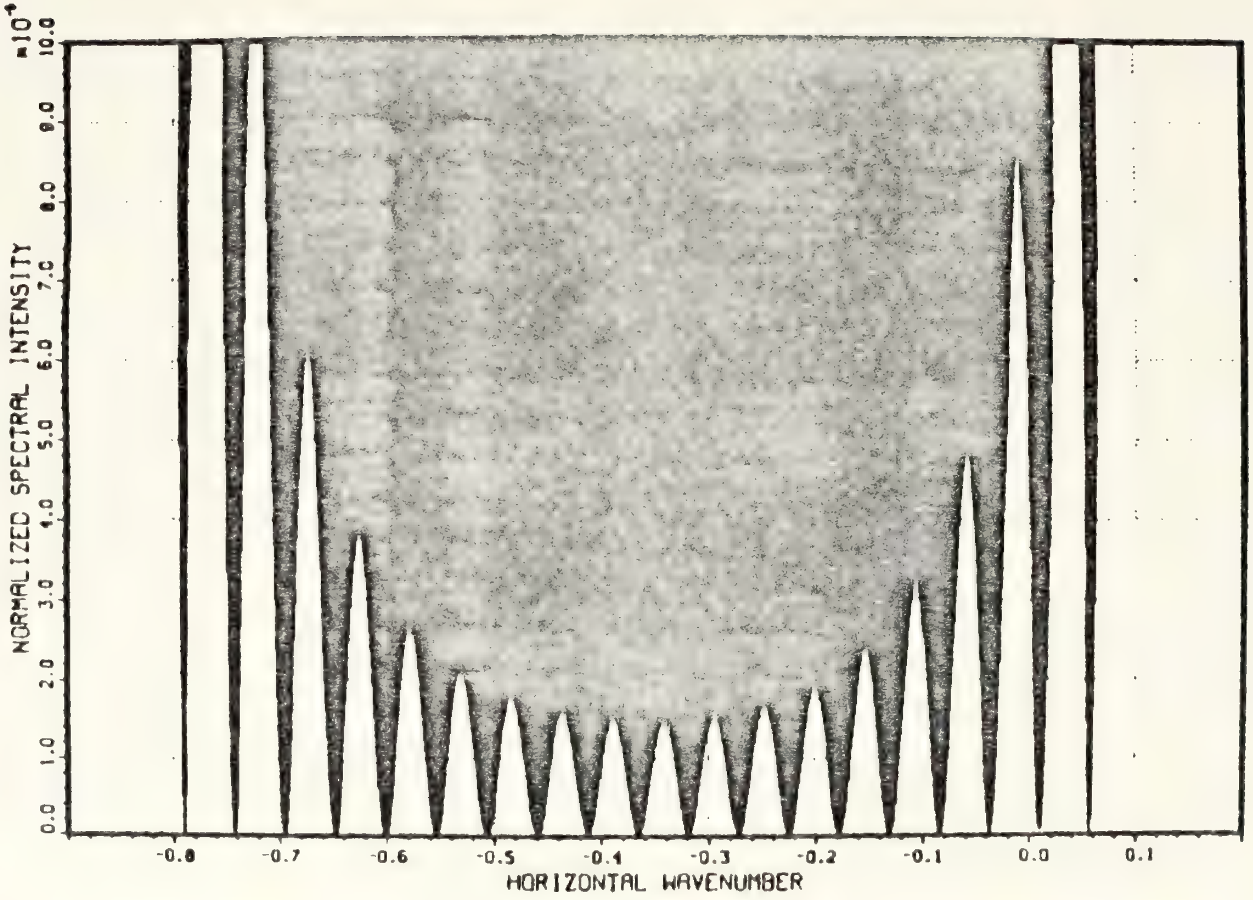

Figure 3.14 IFD: Shallow Water WT, 1 meter Vertical Grid Step.

IV. CONCLUSIONS

Two parabolic equation models, the SSFFT and IFD, were used to predict sound fields for comparison with the Lloyd's mirror interference pattern in the range domain and then in the wavenumber domain. In the range domain, the SSFFT improperly handled the location of pressure nulls and displayed bottom interference. While these weaknesses in the SSFFT were absent in the IFD, the inadvertent and deleterious insertion of noise was extensive. In the wavenumber domain, the noise in the IFD was prohibitive in all but the deep sediment and shallow water cases. The IFD produced results which were comparable with the SSFFT when the noise was not a factor. The range step and the bottom attenuation algorithms of the SSFFT should be reviewed in order to correct those problems described above. In the IFD, the noise problem will require further investigation. If the problem in the IFD is not corrected, this model will be severely limited. With the guidance provided by Lauer [Ref. 2], the wavenumber domain can provide the detailed information needed to quickly correct inconsistencies observed between acoustic models. Therefore, the WT should be investigated further to ensure that any software weaknesses are eliminated and a thorough understanding of the process is assured.

Experiments in the operational environment should be planned to test the WT with actual sound sources. The shipboard experiment could be easily accomplished since the WT can be implemented on micro-computers if the pressure information is provided from an outside source. The WT, if it is going to be considered as a method to determine source depth, appears to have three major shortcomings:

1. A clean CW signal with phase information is essential, which will require quadrature demodulation and extensive filtering.
2. Prediction models are required which will provide equally spaced beta nulls so that software can be developed to automate the process.
3. High resolution processing equipment is necessary because the change in beta spacing for shallow sound sources is small (figure 2.5).

APPENDIX A: SSFFT SOURCE CODE (AT NPS)

```
C
C     PROGRAM PE IS THE INPUT/OUTPUT DRIVER FOR THE PARABOLIC EQUATION
C     MODEL.  IT DEFINES THE FORTRAN UNITS, READS AND PRINTS THE INPUT
C     DATA, CALLS PETL TO CREATE A TRANSMISSION LOSS FILE ON FORTRAN
C     UNIT LT (DISK OR TAPE), CHECKS FOR A CRASH, AND PRINTS A FIELD
C
C
C     INPUT - LC     FORTRAN INPUT UNIT
C             LP     FORTRAN OUTPUT UNIT
C             LT     FORTRAN FILE UNIT (DISK OR TAPE)
C             LWT    FORTRAN OUTPUT UNIT FOR WAVENUMBER TECHNIQUE
C             TITLE  RUN TITLE (TEXT)
C             ND     NUMBER OF OUTPUT DEPTHS (.LE. 20)
C             IFLAT  FLAT BOTTOM FLAG
C             IPRNT  PRINT FLAG
C             NPLT   NUMBER OF LINE PRINTER FIELD PLOT DEPTHS
C             ZS     INPUT DEPTH (FT)
C             F      FREQUENCY (HZ)
C             CO     REFERENCE SOUND SPEED (FT/SEC OR M/SEC)
C             DMAX   MAXIMUM DEPTH ON TRACK (FT)
C             RMAX   MAXIMUM RANGE OF TRACK (NAUTICAL MILES)
C             DR     RANGE STEP (NAUTICAL MILES)
C             CD1    STARTING DEPTH FOR FIELD PLOT
C             CD2    LAST DEPTH FOR FIELD PLOT
C             CLMIN  MINIMUM FIELD PLOT LOSS
C             DCI    FIELD PLOT LOSS INCREMENT
C             D      OUTPUT DEPTH ARRAY
C             NR     NUMBER OF OUTPUT RANGES
C             NWARN  WARNING MESSAGE COUNT FROM PETL
C
C     INPUT DECK - CARD 1  TITLE  80 COLUMNS OF TEXT
C                                FORMAT(20A4)
C
C                  CARD 2 ... ND,IFLAT,IPRNT,NPLT,ISPH
C                         FORMAT(16I5)
C
C                         ND      NUMBER OF OUTPUT DEPTHS
C                         IFLAT   FLAT BOTTOM FLAG
C                         IPRNT   PRINT FLAG
C                         NPLT    NUMBER OF FIELD PLOT DEPTHS
C                         ISPH    SPERICAL EARTH CORRECTION IF .EQ. 0
C                         IPLOT   GRAPHICS PLOT INDICATOR IF .EQ. 0
C                                 NO PLOT, IF .EQ. 1 PLOT
C                         FACTOR  SCALING FACTOR FOR PLOT SIZE
C                         DMIN    MINIMUM DEPTH, USED BY PLOTI
C                                 DEFAULTS TO ZERO
C
C                  CARD 3 ... ZS,F,BEAM,CO,VABSF
```

```
C                           ZS    SOURCE DEPTH(FT)
C                           F     FREQUENCY (HZ)
C                           BEAM  SOURCE BEAM SIZE (DEGREES)
C                           CO    REFERENCE SOUND SPEED (FT/SEC OR M/S)
C                           VABSF VOLUME ATTENUATION FACTOR
C
C                  CARD 4   DMAX,RMAX,DR,CD1,CD2,CLMIN,DCL
C                           FORMAT(7F10.2)
C
C                           DMAX  MAXIMUM DEPTH
C                                 IF BOTTOM LOSS IS PROVIDED AS SOUND
C                                 SPEED PROFILES IN THE BOTTOM THEN
C                                 THIS PARAMETER MUST BE THE DEPTH AT
C                                 THE BOTTOM OF THE DEEPEST BOTTOM
C                                 PROFILE
C                           RMAX  MAXIMUM RANGE
C                           DR    RANGE STEP
C                           CD1   MINIMUM FIELD PLOT DEPTH
C                           CD2   MAXIMUM FIELD PLOT DEPTH
C                           CLMIN MINIMUM FIELD PLOT LOSS
C                           DCL   FIELD PLOT LOSS INCREMENT
C
C                  CARD 5   D(I),I=1,ND
C                           FORMAT(8F10.2)
C
C                           D     OUTPUT DEPTHS
C
C
C             CARD 6  NPROF ... NUMBER OF PROFILES (FORMAT(I5)
C
C        ( REPEAT CARDS 7 AND 8 FOR EACH PROFILE )
C
C                  CARD 7 ... R(I),NPTS(I) ...
C                       RANGE OF I-TH PROFILE (NMI)
C                       NUMBER OF POINTS IN I-TH PROFILE
C                       FORMAT(F10.2,I5)
C                  CARD 8 ...  (Z(J),C(J),J=1,NPTS(I))
C                       THE PROFILE IN DEPTH,SPEED PAIRS. ENGLISH OR
C                       METRIC. FORMAT(8F10.2)
C
C        CARD 9 ... BATHYMETRY
C             IF IFLAT IS 0, SKIP THIS SEGMENT. ELSE, (RANGE,DEPTH)
C             PAIRS, A TOTAL OF (IFLAT) PAIRS. (NMI,FEET)
C             FORMAT(8F10.2)
C
C
C        CARDS 10-12 ... BOTTOM LOSS INFORMATION ...
C
```

```
C             CARD 10 HORIZONTAL RANGE PERIOD FOR RAYS IN THE ABSORBING
C             BOTTOM. (RECALL THAT THE RANGE PERIOD IN THE BOTTOM
C             IS CONSERVED FOR ALL GRAZING ANGLES). SUGGESTED VALUE
C            IS 6000.0 FEET. (FORMAT(F10.2)  IF THIS VALUE IS LE 0.0,
C             THE CODE WILL GIVE THE USUAL FULLY ABSORBING BOTTOM
C             TREATMENT AND NO FURTHER INPUT IS NEEDED.
C
C             CARD 11 ... NBOTM ... NUMBER OF DISTINCT RANGE REGIONS
C             FOR PARTIALLY ABSORBING BOTTOM. IF GTR 0, THEN L(THETA)
C             TABULATIONS WILL BE SPECIFIED FOR EACH REGION AND THE
C             CODE WILL USE AN ANALYTIC INVERSE PARABOLIC PROFILE FORM.
C             OTHERWISE, THE USER MUST SPECIFY THE ATTENUATION ALPHA(Z)
C             ( IN DB/FT) AND THE (TABULATED) FORM OF THE SOUND SPEED
C             PROFILE IN THE BOTTOM.
C                  (NBOTM FORMAT ... (I5)
C
C        IF NBOTM GT 0 ...
C             CARD 12A  RANGE(I),NTHETA(I)        RANGE (NMI) WHERE
C             THE I-TH BOTTOM REGION STARTS,NUMBER OF POINTS IN THE
C             L(THETA) CURVE. FORMAT(F10.2,I5)
C             CARD 12B (THETA(I,J),LOSS(I,J),J=1,NTHETA(I))    (8F10.2)
C             (TABULATED ANGLE-LOSS CURVE  ... INCIDENT ANGLE,LOSS(DB))
C                  **** REPEAT CARDS 12A-B FOR EACH OF NBOTM REGIONS
C
C        IF NBOTM LT 0 ...
C             CARD 12C ... RANGE(I),NALPHA(I)  (F10.2,I5)  RANGE FOR
C             THE I-TH BOTTOM LOSS REGION (NMI) AND NUMBER OF
C             POINTS IN THE TABULATED ATTENUATION FUNCTION ALPHA.
C             CARD 12D ... (Z(I,J),ALPHA(I,J),J=1,NALPHA(I))   (8F10.2)
C             TABULATED ATTENUATION VS DEPTH CURVE. SEE BELOW.
C             CARD 12E ... NPRPT(I)   ...  NUMBER OF POINTS IN I-TH PROF.
C             CARD 12F (ZM(I,J),CM(I,J),J=1,NPRPT(I))    (8F10.2)
C             USER SPECIFIED PROFILE IN THE ABSORBING BOTTOM.
C       *****
C        *****     NOTE ... IN THE USER SPECIFICATION OF ALPHA(Z) AND
C        *****    SOUND SPEED PROFILE IN THE BOTTOM, THE DEPTHS ARE WITH
C        *****     RESPECT TO THE BOTTOM OF THE WATER COLUMN AS ZERO.
C       *****
C             REPEAT CARDS 12C-F FOR EACH OF IABS(NBOTM) REGIONS
C
C        THE BOTTOM LOSS INPUT IS READ IN BY SUBROUTINE PETL
C
C     CONVENTIONS - IFLAT =  0  THE BOTTOM IS FLAT
C                        .NE.0  BATHYMETRY WILL BE SUPPLIED BY A USER
C
C                   IPRNT =  0  DO NOT PRINT TRANSMISSION LOSS TABLE
C                        .NE.0  PRINT OUTPUT TRANSMISSION LOSS TABLE
C
```

```
C                    NPLT  =  0  DO NOT PLOT FIELD.
C                         .GT.0  PLOT FIELD ON LINE PRINTER.
C
C
C                    CO    =  0  TRANSFORM ENVIRONMENT TO REDUCE
C                                PARABOLIC PHASE VELOCITY ERROR.
C                         .NE.0  NO PARABOLIC PHASE VELOCITY CORRECTION.
C                                RUN WITH SPECIFIED REFERENCE SOUND
C                                SPEED.
C
C                    VABSF =  0  VOLUME ATTENUATION IS CONSIDERED
C                         .GT.0  ATTENUATION IGNORED
C
C                    DR    =  0  VARIABLE STEP SIZE RUN.  RANGE STEP
C                                WILL BE DERIVED FROM THE SPLIT-STEP
C                                ALGORITHM TRUNCATION ERROR ESTIMATES.
C                         .GT.0  RUN AT SPECIFIED RANGE STEP.
C
C                                WARNING - FORCING THE STEP SIZE DISABLES
C                                ALL ERROR CHECKS.
C
C     CONSTANTS - FNM  CONVERSION FACTOR FT/NAUTICAL MILE
C
C     SUBROUTINE - PETL (CREATE TRANSMISSION LOSS FILE)
C
C
C
C     STEP ADVANCES THE SOLUTION ONE RANGE STEP USING THE
C     TAPPERT-HARDIN SPLIT-STEP FOURIER ALGORITHM.
C
C     STEP OPERATES IN TWO MODES -
C
C        1. FIXED STEP - THE SOLUTION IS ADVANCED THE SPECIFIED STEP
C                        SIZE.  NO ERROR CHECKS ARE MADE.
C
C        2. VARIABLE STEP - THE INTEGRAL AND OPERATOR TRUNCATION ERROR
C                           ESTIMATES ARE EXAMINED EVERY FIFTH CALL
C                           AND THE RANGE STEP IS ADJUSTED ACCORDINGLY.
C                           THE TRANSFORM IS ALSO CHECKED FOR EVIDENCE
C                           OF ALIASING.
C
C     INPUT  - PR, PI  THE FIELD AT RANGE = R
C              DR      THE CURRENT RANGE STEP
C
C     OUTPUT - PR, PI  THE FIELD AT RANGE = R + DR
C              DR      NEW RANGE STEP (VARIABLE STEP MODE ONLY)
C
C     RETURN CODE - FLAG  =  0.0 SUCCESSFUL ADVANCE.
```

```
C                          .GT. 0.0 COMPUTED STEP SIZE IS SMALL.
C                                   RANGE STEP NOT CHANGED.
C                          .LT. 0.0 EVIDENCE OF ALIASING.
C                                   (FLAG = TRANSFORM ALIASING TEST)
C
C     LOCAL VARIABLES -  AL2    TRANSFORM L2 NORM
C                        AL4    TRANSFORM ALIASING TEST
C                        ARMS   (SIN(RMS ANGLE))**2
C                        RSTEP  COMPUTED RANGE STEP
C
C     TEMPORARY VARIABLES - AMP, FL, TR, TI
C
C     SUBROUTINES - RST  (REAL VECTOR FAST SINE TRANSFORM)
C                   SET  (CONSTRUCT STORED TABLES)
C
C#####################################################################
C#################### START OF EXECUTABLE CODE ######################
      INTEGER TITLE
      REAL LOSSI
      DIMENSION L(20),IBUFO(21)
C
C     CODE BETWEEN $$$$$$$ HAS BEEN ADDED TO ENABLE DISSPLA GRAPHICS
C     AND WAVENUMBER TECHNIQUE FOR NPS INSTALLATION
C
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
      COMMON /PLTG/ IPLOT,FACT,DDDD
      COMMON /WTFLD/WPR(2049),WPI(2049),PRS(2049),GA
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
      COMMON /UNITS/ LC,LP,IT
      COMMON/EARTH/ISPH
      COMMON /OUTBUF/ NOUT,BUFO(21)
      COMMON /HERTZ/ CO,H,HK,F,FK,FACTOR,WL
      COMMON/PLT/TITLE(20),NPLT,LCR,CLMIN,DCL,CD1,CD2,DCD,CD(120)
      COMMON /LOSFCN/ THETI(50),LOSSI(50),NLTH,ZHAT(100),ALGEN(100),
     1 HORRAN,JA,NHAT,RAPRES,RANEXT,CMUD(30),ZMUD(30),NMUD,NBCTM
      REAL LOSSI
      COMMON /MESH/ R,DR,NR,KR,DZ,ZMAX,IB,N,NPTS,N2,N4,NL4,NA,NW,ZW,
     1  HALF,NEFFW
      COMMON / CCSTR / VABSF , ATTEN
      COMMON/ERRCRF/FLAG
      EQUIVALENCE (BUFO(1),IBUFO(1))
      DATA FNM/6076.1/
C
C     DEFINE FORTRAN UNITS.
C
      LC=5
      LP=6
      IT=1
```

```
      LWT=3
      LWTF=7
      REWIND 4
C
C     READ INPUT DATA.
C
      READ(LC,900)TITLE
  900 FORMAT(20A4)
C
      READ(LC,910) ND,IFLAT,IPRNT,NPLT,ISPH,IPLOT,FACT,DMIN,IFLAGU
910   FORMAT(6I5,2F5.0,I5)
      NPLT=MIN0(NPLT,120)
C
      READ(LC,920) ZS,F,BEAM,C0,VABSF
      ZSHOLD = ZS
920   FORMAT(8F10.2)
C
      READ(LC,920)DMAX,RMAX,DR,CD1,CD2,CLMIN,DCL
C
      RRMAX=RMAX
      READ(LC,920) (D(I),I=1,ND)
C
C
      READ(LC,910) NPROF
      IF(NPROF.NE.0) CALL RDPROF(NPROF)
C
      CALL GETBOT(IFLAT,DMAX)
C
      WRITE(LP,930)TITLE
  930 FORMAT(1H1,20A4)
C
      WRITE(LP,935) ND,IFLAT,IPRNT,NPLT,ISPH,IPLOT
935   FORMAT(14HCINPUT DATA --/
     1       13H0PARAMETERS -,16I5)
C
C
      IF (IFLAT.EQ.0) WRITE(LP,950)
950   FORMAT(20H0THE BOTTOM IS FLAT.)
C
      IF (IFLAT.NE.0) WRITE(LP,955)
955   FORMAT(31H0THE BOTTOM IS RANGE DEPENDENT.)
C
C
      WRITE(LP,960) DMAX,ZS,F
  960 FORMAT(17H0MAXIMUM DEPTH =    ,F7.1,3H FT/,
     1 17H SOURCE DEPTH   =   ,F7.1,3H FT,/,
     2 17H FREQUENCY        = ,F7.1,3H HZ)
C
```

```
      IF (BEAM.LT.1.)BEAM=20.
      WRITE(LP,965) BEAM
965   FORMAT(17H BEAM WIDTH     = ,F7.1,8H DEGREES)
C
      WRITE(LP,970)
970   FCRMAT(14HOCUTPUT DEPTHS)
C
      DC 10 I=1,NI
      FI=I
10    WRITE(LP,975)FI,D(I)
975   FORMAT(F4.C,F8.1,3H FT)
C
C        SPHERICAL EARTH CCRRECTION
C        NO CORRECTION IF ISPH GT 0
      IF(ISPH.GT.0) GO TO 12
C
      DC 11 I=1,NI
      D(I)=D(I)*(1.0+D(I)/4.1807E7)
   11 CCNTINUE
      ZS=ZS*(1.0+ZS/4.1807E7)
   12 CCNTINUE
C
C
      IF (VABSF.GT.0.) WRITE(LP,979)
979   FCRMAT(36HOVOLUME ATTENUATION HAS BEEN IGNCRED)
C
C
      IF (DR) 16,17,15
C
15    WRITE(LP,980)DR
980   FORMAT(14HORANGE STEP = ,F5.3,4H NM.)
      DRHOLD = DR
      DR=FNM*DR
      GC TO 18
C
16    DR=0.
17    WRITE(LP,985)
      DRHOLD = DR
985   FCRMAT(24HOVARIABLE STEP SIZE RUN.)
C
   18 CONTINUE
      RMAX=FNM*RMAX
C
C     CCMPUTE TRANSMISSICN LOSS.
C
      REWIND LT
C
      CALL PETL(ZS,BEAM,NL,D,DMAX,RMAX,IFLAT,NWARN)
```

```
C
      ENDFILE LT
      REWIND LT
C
      WRITE(LP,995)NR
995   FORMAT(30H0RUN TERMINATED AT RANGE STEP ,I5,1H.)
C
C     CHECK FOR CRASH.
C
      IF (NWARN.GE.5) WRITE(LP,9900)
9900  FORMAT(1H0,6X,45(1H/),4X,23HWARNING LIMIT EXCEEDED.,4X,45(1H/)/
     1        54X,28HTHE PROGRAM CANNOT CONTINUE.)
C     IF(NWARN.GE.5) GO TO 40
      IF(FLAG.LT.-14.) GO TO 40
C
C
C     CODE BETWEEN $$$$$$$ HAS BEEN DISABLED TO ENABLE DISSPLA
C     GRAPHICS FOR NPS INSTALLATION
C
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
CCCC  IF(IPLOT.NE.0) CALL PLOTL(FACT,RRMAX,DMIN)
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
      REWIND LT
C
      IF (IPRNT.EQ.0) GO TO 31
C
C     PRINT TRANSMISSION LOSS TABLE.
C
      WRITE(LP,930)TITLE
C
      READ(LT) X
C
      DO 20 I=1,ND
 20   IBUFO(I)=D(I)+0.50
C
      WRITE(LP,9910)ZS,F,(IBUFO(I),I=1,ND)
9910  FORMAT(21H0TRANSMISSION LOSS --/
     1 16H SOURCE DEPTH = ,F6.1,3H FT/,
     2 16H FREQUENCY =              ,F6.1,3H HZ    ,/,
     3       8H0DEPTH =,20I6)
C
      WRITE(LP,9915)
9915  FORMAT(8H RANGE   ,120(1H-))
C
C
C
C     CODE BETWEEN $$$$$$$ HAS BEEN ADDED TO ENABLE DISSPLA GRAPHICS
C     FOR NPS INSTALLATION AND WAVENUMBER TECHNIQUE
```

```
C
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
   31 CONTINUE
      WRITE(LWT,900)TITLE
      WRITE(LWT,32)  F,ZSHOLD,NR,ND,CLMIN,DCL,FACT,IPLOT
   32 FORMAT(2(2X,F10.2),2(2X,I4),3(2X,F7.2),2X,I3)
      WRITE(LWT,33)(D(I),I=1,ND)
   33 FORMAT(5E15.7)
      WRITE(LWT,34)DDDD,FK,CO,ATTEN,DMAX
   34 FORMAT(5(1X,E15.7))
      WRITE(LWT,39)BEAM,DRHOLD,IFLAGU
   39 FORMAT(2(1X,F10.2),1X,I1)
      IF (IPRNT.EQ.0) READ(LT) X
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
      DO 30  I=1,NR
      READ(LT) (BUFO(J),J=1,NOUT)
C
C     CODE BETWEEN $$$$$$$ HAS BEEN ADDED TO ENABLE DISSPLA GRAPHICS
C     FOR NPS INSTALLATION AND WAVENUMBER TECHNIQUE
C
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
      WRITE(LWT,33)(BUFO(IWT1),IWT1=1,NOUT1)
      WRITE(LWTF,35)  BUFO(1),WPR(I),WPI(I)
   35 FORMAT(3(2X,E15.7))
      IF (IPRNT.NE.0) WRITE(LP,9920)(BUFO(J),J=1,NOUT)
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
      WRITE(LP,9920) (BUFO(J),J=1,NOUT)
   30 CONTINUE
9920  FORMAT(F8.2,20F6.1)
C
40    STOP
      END
C     FETL IS THE MAIN SUBROUTINE OF THE PARABOLIC EQUATION MODEL.
C     IT DEFINES CONSTANTS, CONTROLS THE RANGE LOOP, AND CREATES AN
C     UNFORMATTED OUTPUT FILE (DISK OR TAPE) CONTAINING RANGES AND
C     THE ASSOCIATED TRANSMISSION LOSS AT SPECIFIED DEPTHS.
C
C     INPUT  - LP     FORTRAN OUTPUT UNIT (PRINTER)
C              LT     FORTRAN OUTPUT UNIT (DISK OR TAPE)
C              RMAX   MAXIMUM RANGE OF CALCULATION (FT)
C              DMAX   MAXIMUM DEPTH (FT)
C              ND     NUMBER OF OUTPUT DEPTHS (.IE. 20)
C              D      OUTPUT DEPTH ARRAY (ND DEPTHS)
C              NPLT   NUMBER OF FIELD PLOT DEPTHS
C              CD1    MINIMUM FIELD PLOT DEPTH
C              CD2    MAXIMUM FIELD PLOT DEPTH
C              ZS     INPUT DEPTH (FT)
C              IFLAT  FLAT BOTTOM FLAG (0 FOR FLAT BOTTOM)
```

```
C             DR     CURRENT RANGE STEP (FT)
C             F      FREQUENCY (HERTZ)
C             NC     NUMBER OF POINTS ON THE SOUND VELOCITY PROFILE
C             Z      DEPTH ARRAY (NC DEPTHS)
C             C      SOUND SPEED ARRAY (NC SOUND SPEEDS)
C             RNEXT  RANGE OF NEXT SOUND VELOCITY PROFILE (FT)
C             FLAG   INTEGRATION STATUS FLAG FROM STEP
C
C      OUTPUT - NWARN  WARNING MESSAGE COUNT
C             ZMAX   MAXIMUM DEPTH SAMPLE IN THE TRANSFORM
C             DCL    FIELD PLOT DEPTH INCREMENT
C             CL     FIELD PLOT DEPTHS
C             DM     TRANSFORMED OUTPUT DEPTHS
C             CC     REFERENCE SOUND SPEED (FT/SEC)
C             MC     PHASE VELOCITY CORRECTION FLAG
C             WL     ACOUSTIC WAVELENGTH (FT)
C             FK     AVERAGE WAVE NUMBER
C             H      MESH INCREMENT IN TRANSFORM SPACE
C             HK     RATIO OF MESH INCREMENT IN TRANSFORM SPACE
C                    TO AVERAGE WAVE NUMBER
C             NPTS   NUMBER OF POINTS IN DEPTH MESH (NPTS = 2**N - 1)
C             HALF   HALF THE NUMBER OF DEPTH MESH POINTS
C             DZ     DEPTH MESH INCREMENT (FT)
C             NW     NUMBER OF MESH POINTS IN THE WATER COLUMN
C             NA     NUMBER OF MESH POINTS IN THE ABSORBING LAYER
C             R      CURRENT RANGE (FT)
C             NR     CURRENT RANGE STEP COUNT
C             ZW     BOTTOM DEPTH AT CURRENT RANGE
C             IB     MESH INDEX OF BOTTOM
C             LCR    LAST PRINT PLOT RANGE
C      FILE OUTPUT
C             RNM    CURRENT RANGE (NAUTICAL MILES)
C             TL     TRANSMISSION LOSS (DB)
C
C      LOCAL VARIABLES - RR  RECIPROCAL RANGE
C                        NMAX  MAXIMUM TRANSFORM SIZE
C
C      CONSTANTS - FNM  CONVERSION FACTOR FT/NAUTICAL MILE
C
C      SUBROUTINES - FILTER (INTERPOLATE AND SMOOTH INDEX OF REFRACTION
C                            ON THE FIELD MESH)
C                    INDEX  (CONSTRUCT INDEX OF REFRACTION TABLE)
C                    SPEED  (EVALUATE SOUND SPEED AT SPECIFIED DEPTH)
C                    SOURCE (GENERATE INITIAL FIELD)
C                    SET    (CONSTRUCT STORED TABLES)
C                    STEP   (SPLIT-STEP FOURIER INTEGRATION ALGORITHM)
C                    TLOSS   (INTERPOLATE FIELD AND RETURN
C                             TRANSMISSION LOSS)
```

```
C                   FLD       (FIELD PRINT PLOTTER)
      SUBROUTINE FETL(ZS,BEAM,ND,D,DMAX,RMAX,IFLAT,NWARN)
      INTEGER TITLE
      DIMENSION L(20),BUFO(21)
C
C     CODE BETWEEN $$$$$$$ HAS BEEN ADDED TO ENABLE DISSPLA GRAPHICS
C     FOR NPS INSTALLATION AND WAVENUMBER TECHNIQUE
C
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
      COMMON /PLTG/ IPLOT,FACT,DDDD
C     DIMENSION CDOWN(5000)
      COMMON /WTFLD/WPR(2049),WPI(2049),PRS(2049),GA
      COMMON /FIELD/ PR(2049),PI(2049)
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
      COMMON /UNITS/ LC,LP,LT
      COMMON /LOSFCN/ THETI(50),LOSSI(50),NLTH,ZHAT(100),ALGEN(100),
     1 HORRAN,JA,NHAT,RAPRES,RANEXT,CMUD(30),ZMUD(30),NMUD,NBOTM
      REAL LOSSI
      COMMON /OUTBUF/ NOUT,RNM,TL(20)
      COMMON /HERTZ/ CO,H,HK,F,FK,FACTOR,WL
      COMMON /PHASE/ NC,Z(100),C(100),MC,DM(20)
      COMMON/PLT/TITLE(20),NPLT,LCR,CLMIN,DCL,CD1,CD2,DCD,CD(120)
      COMMON /BATHY/ RE,KE,NB,BR(101),BZ(101)
      COMMON/ERRCRF/FLAG
C
      COMMON /MESH/ R , DR , NR , KR , DZ , ZMAX , IB , N , NPTS , N2 ,
     A              N4 , NL4 , NA , NW , ZW , HALF , NEFFW
C
      COMMON / COSTR/ VABSF,ATTEN
      EQUIVALENCE (BUFO(1),RNM)
      DATA FT,FNM,TWOPI,CUT/0.3048,6076.1,6.28318530717959,-14.0/
      DATA RAD/0.17453292519943E-01/
C
C     DEFINE MAXIMUM TRANSFORM SIZE.
C
      NMIN=5
      NMAX=2048
C
C        EXTEND THE PHYSICAL EXTENT OF THE PROBLEM TO 4/3 THE
C             EFFECTIVE WATER COLUMN DEPTH, WHERE THE EFFECTIVE
C             WATER COLUMN IS GIVEN BY THE ACTUAL WATER COLUMN PLUS
C             TWENTY PERCENT OF THE HORIZONTAL RANGE PERIOD. THIS
C             SHOULD BE SUFFICIENT TO TURN AROUND RAYS OF UP TO 38 DEGRE
C             IN THE ELLIPTIC APPROXIMATION.
      READ(LC,702) HORRAN
      HORRAN=AMAX1(HORRAN,0.0)
      ZMAX = (4.0/3.0) * (DMAX + (HORRAN/5.0) )
C
```

```
C     SET THE FIELD PLOT DEPTH INCREMENT.
C
      IF (NPLT.GT.0) DCD=AINT((CD2-CD1)/FLOAT(NPLT-1))
C
C     SET THE PHASE VELOCITY CORRECTION FLAG.
C
      REWIND 2
C
      IF (C0.LE.0.) GO TO 5
C
C     THE REFERENCE SOUND SPEED HAS BEEN SPECIFIED.
C     DO NOT TRANSFORM THE ENVIRONMENT TO REDUCE THE PARABOLIC PHASE
C     VELOCITY ERROR.  SET THE FLAG AND THE TRANSFORMED OUTPUT DEPTHS.
C
      MC=2
C
      READ(2)
C
      DO 1 I=1,ND
1     DM(I)=D(I)
      IF (NPLT.LE.0) GO TO 8
      CD(1)=CD1
      DO 2 I=2,NPLT
2     CD(I)=CD(I-1)+DCD
      GO TO 8
C
C     THE REFERENCE SOUND SPEED HAS NOT BEEN SPECIFIED.
C     THE ENVIRONMENT WILL BE TRANSFORMED TO REDUCE THE PARABOLIC
C     PHASE VELOCITY ERROR.  SET THE FLAG AND DEFINE THE REFERENCE
C     SOUND SPEED.
C
5     MC=1
C
      READ(2)C0
C
8     IF (C0.LE.3000.) C0=C0/FT
C
C                DEFINE THE VOLUME ATTENUATION FACTORS
C
      CONV=2.302585/(20.*FNM)
      FKHZ=F*.001
      FKHZ2=FKHZ**2
      IF(FKHZ.GT.1.) GO TO 888
      ATTEN=.125*FKHZ2*CONV
      GO TO 889
888   CONTINUE
      ATTEN=2.*FKHZ2*(.1/(1.+FKHZ2)+40./(4100.+FKHZ2))*CONV
889   CONTINUE
```

```
C
C
C
C
C     DEFINE THE AVERAGE ACOUSTIC WAVELENGTH AND WAVE NUMBER
C     AND THE MESH INCREMENT IN TRANSFORM SPACE.
C
      WI=CO/F
      FK=TWOPI/WI
      H=TWOPI/(ZMAX+ZMAX)
      HK=H/FK
C
C
C                         --TRANSFORM SIZE--
C                   DETERMINE THE NUMBER OF POINTS REQUIRED
C
C
      STHC=SIN(RAD*BEAM)
C
      N=1.+ALOG(4.*STHC/(3.*HK))/ALOG(2.0)
C
      IF (N.LE.NMIN) N=NMIN
      WRITE(LP,900) N
900   FORMAT(30H0SELECTED TRANSFORM SIZE = 2**,I2)
C
C     DEFINE CONSTANTS.
C
10    NPTS=2**N
C
C     DETERMINE EFFECTIVE BEAM WIDTH
C
      STHCP=(3./4.)*NPTS*HK
      IF(STHCP.GT.1.0)  STHCP=1.0
      THC=ARSIN(STHCP)/RAD
C
      WRITE(LP,902) THC
  902 FORMAT(' EFFECTIVE BEAM WIDTH (DEG) = ',F4.1)
C
C
C     CHECK TRANSFORM SIZE.
C
      IF (NPTS.LE.NMAX) GO TO 15
C
      WRITE(LP,905)
905   FORMAT(41H0TRANSFORM SIZE EXCEEDS ARRAY DIMENSICNS.)
      STOP
C
15    N2=NPTS/2
```

```
      N4=N2/2
      NI4=NPTS-N4
      HALF=N2
      DZ=ZMAX/FLOAT(NPTS)
      NPTS=NPTS-1
      NW = DMAX/DZ + 0.5
      NOUT = ND + 1
      NBOTM = 0
      NEFFW = (DMAX + 0.2*HORRAN)/DZ + 0.5
      NA=NPTS-NEFFW
      IF(HORRAN.LE.0.0) GO TO 735
C**********************************************************************
C
C         EFFECTIVE WATER COLUMN IS DMAX + 20 PERCENT OF HORIZONTAL
C         RANGE PERIOD.
C         NEFFW IS THE INDEX OF THE **EFFECTIVE WATER DEPTH**
C         NW RETAINS THE CUSTOMARY DEFINITION
C         NA IS THE NUMBER OF POINTS IN THE FULLY ABSORBING BOTTOM.
C         GENERATE OR READ BOTTOM ATTENUATION FUNCTION
      READ(LC,701) NBOTM
  701 FORMAT(I5)
C         KEY ON NBOTM GTR 0 FOR L(THETA,R), LT 0 FOR C,ALPHA(Z)
      IF(NBOTM.LT.0) GO TO 720
C         USER SPECIFIES L(THETA,R)
      DO 710 I=1,NBOTM
C         GET APPLICABLE RANGE AND NUMBER OF POINTS
      READ(LC,702) R,NLTH
  702 FORMAT(F10.2,I5)
      READ(LC,703) (THETI(L),LOSSI(L),L=1,NLTH)
  703 FORMAT(8F10.2)
      WRITE(LP,704) R,HORRAN
  704 FORMAT(//,25H LOSS FUNCTION PARAMETERS    ,/,
     A   19H RANGE,RANGE PERIOD     ,2F10.2,/,
     B   5X,5HANGLE,2X,8HLOSS(DB)         )
      WRITE(LP,705) (THETI(L),LOSSI(L),L=1,NLTH)
  705 FORMAT(2F10.2)
C         GENERATE ALPHA(Z)
      CALL LOSGEN
C         DISPLAY FUNCTION
      WRITE(LP,706)
  706 FORMAT(/,12H ALPHA VS Z    ,/,9X,1HZ,2X,8HALPHA(Z)     )
      WRITE(LP,707) (ZHAT(I),ALGEN(L),L=1,NHAT)
  707 FORMAT(F10.3,F10.6)
      BACKSPACE 4
      WRITE(4) R,NHAT,ZHAT,ALGEN
      R1=1.0E15
      R=R*FNM
      WRITE(4) R1
```

```
  710 CONTINUE
      GO TO 735
C
C        USER SPECIFIED C(Z)  AND  ALPHA(Z)
C
  720 NNN=IABS(NBOTM)
      DO 730 I=1,NNN
      READ(LC,702) R,NHAT
      READ(LC,703)  (ZHAT(L),ALGEN(L),L=1,NHAT)
      READ(LC,701)  NMUD
      READ(LC,703)  (ZMUD(L),CMUD(L),L=1,NMUD)
      BACKSPACE 4
      R=R*FNM
      WRITE(4)  R,NHAT,ZHAT,ALGEN,NMUD,ZMUD,CMUD
      R1=1.0E15
      WRITE(4)  R1
      WRITE(LP,706)
      WRITE(LP,707)  (ZHAT(I),ALGEN(L),L=1,NHAT)
  730 CONTINUE
  735 CONTINUE
      REWIND 4
C**********************************************************************
C
C
C     INITIALIZE FOR RANGE LOOP.
C
      R=0.
      KB=0
      NR=0
      KR=1
      LCR=0
      NWARN=0
      IB=NPTS
      RE=0.0
      RANEXT=1.0E10
      READ(2)
17    CALL SVP(NC,Z,C,RNEXT)
      IF (RE.GT.RNEXT) GO TO 17
   18 CONTINUE
      IF(NBOTM.EQ.0)  GO TO 19
      IF(NBOTM.GT.0)  READ(4)  RAPRES,NHAT,ZHAT,ALGEN
      IF(NBOTM.LT.0)  READ(4)  RAPRES,NHAT,ZHAT,ALGEN,NMUD,ZMUD,CMUD
      READ(4) RANEXT
      BACKSPACE 4
      IF(RE.GT.RANEXT)  GO TO 18
   19 CONTINUE
      CALL FILTER(ND,D)
      DATE = 1078.
```

```
C
      WRITE(LT)TITLE,WHEN,F,ZS,NR,ND,(D(I),I=1,ND)
C     CONSTRUCT THE INITIAL FIELD.
C
      ZSM=ZS
      IF (MC.EQ.1) ZSM=ZSM*SQRT(C0/SPEED(ZSM))
      CALL SOURCE(ZSM)
C
C
C        INTRODUCE BOTTOM LOSS TO FILTER FUNCTION IF APPLICABLE
C
      IF(NBOTM.NE.0) CALL FILLOS
C     IF THE RANGE STEP HAS BEEN SPECIFIED, CONSTRUCT THE STORED
C     TABLES AND TURN OFF THE ERROR CHECKS BY ZEROING THE FLAGGED
C     STEP COUNTER.
C
      IF(DR.GT.0.0) KR=0
C
C     CODE BETWEEN $$$$$$$ HAS BEEN ADDED TO ENABLE DISSPLA GRAPHICS
C     FOR NPS INSTALLATION AND WAVENUMBER TECHNIQUE
C
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
      DO 92 IWT = 1,NW
      DDDD = IWT * DZ
      IF(ZS .LT. DDDD) GO TO 93
   92 CONTINUE
   93 CONTINUE
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C
C     BEGIN RANGE LOOP.
C
20    NR=NR+1
C
C     GET BOTTOM INDEX AT CURRENT RANGE.
C
      IF (IFLAT.EQ.0) GO TO 30
C
      ZW=ZB(RE)
      IF (MC.EQ.1) ZW=ZW*SQRT(C0/SPEED(ZW))
      IB = ZW/DZ + 0.5
      IF(IB.GE.NW) IB = NPTS
C
C     ADVANCE THE SOLUTION.
C
30    CALL STEP(FLAG)
C
      R=R+DR
      RE=R
```

```
      RNM=R/FNM
      RR=9.0/R
C
C     INTERPOLATE FOR TRANSMISSION LOSS AT OUTPUT DEPTHS.
C
      DO 40  I=1,ND
40    TL(I)=TLOSS(RR,DM(I))
C
C     CODE BETWEEN $$$$$$$ HAS BEEN ADDED TO ENABLE DISSPLA GRAPHICS
C     FOR NPS INSTALLATION AND WAVENUMBER TECHNIQUE
C
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
      WER(NR) = ER(IWT)
      WEI(NR) = EI(IWT)
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C
C     DUMP RANGE AND TRANSMISSION LOSS TO OUTPUT TAPE.
C
      WRITE(LT) (BUFO(I),I=1,NOUT)
C
C     PRINT FIELD PLOT IF FLAGGED.
C
      IF (NPLT.GT.0) CALL FLD(RR)
C
C     CHECK FOR NEW VELOCITY PROFILE.
      IF(RE.LT.RNEXT .AND. RE .LT.RANEXT)  GO TO 60
      IF(RE.LT.RNEXT)  GO TO 52
   50 CALL SVP(NC,Z,C,RNEXT)
      IF(RE.GE.RNEXT)  GO TO 50
   52 IF(RE.LT.RANEXT)  GO TO 56
   54 CONTINUE
      IF(NBOTM.EQ.0)  GO TO 56
      IF(NBOTM.GT.0)  READ(4)  KAPRES,NHAT,ZHAT,ALGEN
      IF(NBOTM.LT.0)  READ(4)  KAPRES,NHAT,ZHAT,ALGEN,NMUD,ZMUD,CMUD
      READ(4) RANEXT
      BACKSPACE 4
      IF(RE.GE.RANEXT)  GO TO 54
      CALL FILLOS
   56 CONTINUE
      CALL FILTER(ND,D)
      CALL INDEX
C
60    IF (FLAG) 80,90,70
C
C     PROCESS AN ERROR RETURN FROM STEP.
C
70    WRITE(LP,910) RNM
910   FORMAT(26H0****WARNING - AT RANGE = ,F7.2,4H NM,)
```

```
C
C      COMPUTED RANGE STEP IS SMALL.
C
       WRITE(LP,920)FLAG
920    FORMAT(15X,32HORATIO OF COMPUTED RANGE STEP TO,
      1            22H ACOUSTIC WAVELENGTH =,E10.3)
C
       NWARN=NWARN+1
C
       IF (NWARN.GE.5) GO TO 100
C
       WRITE(LP,930)
930    FORMAT(15X,33HPROCEEDING AT CURRENT RANGE STEP.)
       GO TO 90
C
C      TRANSFORM ALIASING TEST FAILED.
C
80     WRITE(LP,910)RNM
       WRITE(LP,940)FLAG
940    FORMAT(15X,25HTRANSFORM ALIASING TEST =,F6.1,4H DB.)
C
       NWARN=NWARN+1
C
C      TERMINATE THE RUN IF ALIASING IS SEVERE.
C
       IF (FLAG.GT.CUT)  NWARN=NWARN+4
C
       IF (NWARN.GE.5) GO TO 100
C
C      END RANGE LOOP.
C
90     IF (R.LT.RMAX) GO TO 20
C
100    RETURN
       END
C
       SUBROUTINE STEP(FLAG)
       COMMON /LOSFCN/ THETI(50),LOSSI(50),NLTH,ZHAT(100),ALGEN(100),
      1 HORRAN,JA,NHAT,RAPRES,RANEXT,CMUD(30),ZMUD(30),NMUD,NBCTM
       REAL LOSSI
       COMMON /FIELD/ PR(2049),PI(2049)
       COMMON /HERTZ/ CO,H,HK,F,FK,FACTOR,WL
       COMMON /BATHY/ RE,KE,NB,BR(101),BZ(101)
C
       COMMON /MESH/ R,DR,NR,KR,DZ,ZMAX,IB,N,NPTS,N2,N4,NL4,NA,NW,ZW,
      1  HALF,NEFFW
C
       COMMON /TABLE/ SR(2047),SI(2047),UR(2047),UI(2047),FN(2047)
```

```
      COMMON /OUTBUF/NOUT,RNM,TL(20)
      COMMON /COSTR/ VABSF,ATTEN
      COMMON/FIL1/FIL(2047)
      DATA   ALIAS,DRMAX,RMSMIN/1.0E-02, 3038.05, 1.0E-02/
      DATA NCB,NBEF/0, 0/
      DATA NOB/0/,NBEF/0/
C
C
C     SET RETURN FLAG.
C
      FLAG=0.0
C
C
C        FOURIER TRANSFORM (ONLY IF NOT THE FIRST STEP)
C
      IF(NR.EQ.1) GO TO 1
C
      CALL RST(PR,N,1)
      CALL RST(PI,N,1)
   1  CONTINUE
C
C        CHECK FOR FIRST STEP OF FIXED-STEP -- APPLY SPLIT-STEP
C
      IF(KR.NE.0) GO TO 3
      IF(NR.NE.1) GO TO 60
      RSTEP = DR
      DR = 0.0
      GO TO 40
    3 CONTINUE
C
C        COMPUTE VARIABLE RANGE STEP SIZE BY FIRST
C        SEARCHING FOR K-SPACE PEAK
C
      FL = 0.
      AL2 = 0.
      ARMS = 0.
      APEAK = 0.
C
      DO 5 I = 1,NPTS
      AMP = PR(I)*PR(I)  + PI(I)*PI(I)
      AL2 = AL2 + AMP
      APEAK = AMAX1(APEAK,AMP)
C
      FL = FL + EK
      ARMS = ARMS + FL*FL*AMP
    5 CONTINUE
      ARMS = ARMS/(AL2+ARMS)
C
C        FIND LAST POINT IN K-SPACE AT LEAST
```

```
C         50 DB DOWN FROM PEAK
C
      N50 = NPTS
      DO 8 I = 1,NPTS
      AMP = PR(N50)*PR(N50) + PI(N50)*PI(N50)
      RATIO = AMP/APEAK
      IF(RATIO.GE.1.E-5) GO TO 10
      N50 = N50 - 1
    8 CONTINUE
C
C         DETERMINE RANGE STEP USING 50 DB DOWN ANGLE
C         (COMPUTE RMS ANGLE FOR COMPARISON)
C
   10 SINA = FLOAT(N50-1)*HK
      SINA2 = SINA*SINA
      IF(SINA2.GT.1.0) SINA2 = 1.0
      COSA = SQRT(1.-SINA2)
      RSTEP = AMIN1(WL/(1.-COSA),DRMAX)
C
C         CHECK FOR DIAGNOSTIC STEP
C
      IF(NR-KR) 30,15,60
C
C         DIAGNOSTIC STEP - EXECUTE ALIASING TEST, EXAMINE
C         ENERGY DISTRIBUTION IN K-SPACE
C
   15 KR = KR + 5
      HI2 = 0.
      HI4 = 0.
      NL16 = NPTS - NPTS/16
      AL16 = 0.
      DO 20 I = 1,NPTS
      AMP = PR(I)*PR(I) + PI(I)*PI(I)
      IF(I.LE.N2) HL2 = HL2 + AMP
      IF(I.LE.N4) HL4 = HI4 + AMP
      IF(I.GE.NL16) AL16 = AL16 + AMP
   20 CONTINUE
C
C         SET ALIASING FLAG IF TOLERANCE EXCEEDED
C
      AL16 = AL16/AL2
      IF(AL16.GT.ALIAS) FLAG = 10.*ALOG10(AL16)
C
      IF(DR.LE.0.) GO TO 40
C
C         TRUNCATE SPECTRUM IF POSSIBLE
C
      IF((ARMS.LT.RMSMIN).OR.(NBOTM.NE.0)) GO TO 30
```

```
C
      HL4 =  (HL2-HL4)/HL4
      HL2 =  (AL2-HL2)/HL2
      IF((HL2.GT.1.E-7) .OR.(HL4.GT.1.E-6)) GO TO 30
C
C     TRUNCATE THE SPECTRUM.
C
      N=N-1
      NPTS=NPTS/2
      N2=N2/2
      N4=N4/2
      NL4=NL4/2
      IB=IB/2
      NA=NA/2
      NW=NW/2
      NEFFW=NEFFW/2
      DZ=DZ+DZ
      HALF=0.50*HALF
C
      K=0
      DO 25 I = 1,NPTS
      K=K+2
      FIL(I)=FIL(K)
      FN(I)=FN(K)
      UR(I)=UR(K)
      UI(I)=UI(K)
      AMP=SR(I)*PR(I)-SI(I)*PI(I)
      PI(I)=SR(I)*PI(I)+SI(I)*PR(I)
      PR(I)=AMP
      SR(I)=SR(I)+SR(I)
   25 SI(I)  = SI(I)  + SI(I)
C
      GO TO 80
C
C
C     CHECK RELATIVE CHANGE IN STEP SIZE.
C
30    IF (ABS(DR-RSTEP)/DR.LE.0.25)  GO TO 60
C
C     PREPARE FOR NEW RANGE STEP.
C
40    FACTOR=0.25*(DR+RSTEP)/FK
      DR=RSTEP
      NFIL = (3./4.) * FLOAT(NPTS+1)
C
      DO 50 I=1,NPTS
      FL=H*FLOAT(I)
      FI=FACTOR*FI*FL
```

```
      TR=COS(FL)/HALF
      IF(I.GT.NFIL) TR=FIL(I)*TR
      TI=-SIN(FL)/HALF
      IF(I.GT.NFIL) TI=FIL(I)*TI
      AMP=TR*PR(I)-TI*PI(I)
      PI(I)=TR*PI(I)+TI*PR(I)
50    PR(I)=AMP
C
C     CONSTRUCT TABLES FOR NEW RANGE STEP.
C
      CALL SET
C
      GO TO 80
C
C     THE RANGE STEP HAS NOT BEEN CHANGED.
C     MULTIPLY BY STORED SECOND DERIVATIVE TRANSFORM TABLE.
C
60    DO 70 I=1,NPTS
      AMP=SR(I)*PR(I)-SI(I)*PI(I)
      PI(I)=SR(I)*PI(I)+SI(I)*PR(I)
70    PR(I)=AMP
C
C     FOURIER TRANSFORM.
C
80    CONTINUE
      CALL RST(PR,N,1)
      CALL RST(PI,N,1)
C
C     CHECK FOR FLAT BOTTOM.
C
      IF(IB-NW) 82,82,100
C
C     THE BOTTOM IS NOT FLAT.
C     SMOOTH THE TRANSITION INTO AN ISOVELOCITY REGION AND
C     PUT IN THE ARTIFICIAL ABSORBING LAYER.
C
82    K=1
      I=IB + 1
C
      L = 1
C
C        I IS ABSOLUTE DEPTH INDEX
C        K IS MESH-POINTS-IN-BOTTOM INDEX
C        INX IS INDEX INTO TABULATED ATTENUATION FUNCTION
C        L IS INDEX INTO FILTER FUNCTION ROLL-OFF FACTOR
C        IB IS INDEX OF BOTTOM AT THIS RANGE
C        NOTE THAT VOLUME ATTENUATION IS CARRIED INTO THE MUD
C
```

```
C        GET VOLUME ATTENUATION FACTOR
C
      ALPHA = 1.0
      IF(VABSF .EQ. 0.0) ALPHA = EXP(-ATTEN*DR)
C
C        GET N**2 - 1.0 AT THE WATER-MUD INTERFACE
C
      NMD=0
      FN0=FN(IB)
      T=FACTOR*FN0
      UREAL=ALPHA*COS(T)*FIL(NW)
      UIMAG=ALPHA*SIN(T)*FIL(NW)
      IF(NBOTM.EQ.0) GO TO 92
C
      F1=3.1415926535/HORRAN
      IF(NBOTM.LT.0) FN1=FNMUD(FN0,0)
      ZM=0.0
      NMD=NEFFW-NW
C
C
C        START LOOP
C
   85 CONTINUE
      IF(NBOTM.LT.0) GO TO 86
C
C        USE ANALYTIC PROFILE IN MUD
C
      ZM=ZM+DZ
      CORR=ZM*F1
      ANMUD=FN0-CORR*CORR
      GO TO 90
C
C        USE USER-SPECIFIED PROFILE
C
   86 FN2=FNMUD(FN1,1)
      ANMUD=0.25*(FN0+FN1+FN1+FN2)
      FN0=ANMUD
      FN1=FN2
C
   90 T=FACTOR*ANMUD
      NNN=NW+K
      XYZ=ALPHA*FIL(NNN)
      UREAL=XYZ*COS(T)
      UIMAG=XYZ*SIN(T)
C
C        CONSTRUCT U*P     (REMEMBER,THEY ARE BOTH COMPLEX)
C
   92 CONTINUE
```

```
      F3 = PR(I)*UREAL - PI(I)*UIMAG
      PI(I)  = PR(I)*UIMAG + PI(I)*UREAL
      PR(I)  = F3
      I = I + 1
      K = K + 1
      IF(I.GT.NPTS) GO TO 94
C        CHECK TC SEE IF WE SHOULD ENTER ISOSPEED EXTENSICN MCDE
      IF(K.LE.NMD) GO TO 85
C        NOW IN FULLY ABSORBING BOTTOM ... CUT IN FILTER FACTCR
      JJJ=NEFFW+I
      IF(JJJ.GT.NPTS) GO TC 93
      PR(I)=PR(I)*FIL(JJJ)
      PI(I)=PI(I)*FIL(JJJ)
      L=L + 1
      GC TO 92
   93 PR(I)=0.0
      PI(I)=0.0
      GC TO 92
   94 CONTINUE
C
C
C     MULTIPLY BY STORED INDEX OF REFRACTION TABLE.
C
100   DC 110 I=1,IB
      AMP=UR(I)*PR(I)-UI(I)*PI(I)
      PI(I)=UR(I)*PI(I) +UI(I)*PR(I)
110   PR(I)=AMP
C
      RETURN
      END
C
C     SET CONSTRUCTS ALL TABLES THAT ARE A FUNCTICN OF THE
C     RANGE STEP (DR).
C
C     INPUT  - DR  THE CURRENT RANGE STEP
C
C     OUTPUT - A   THE ARTIFICIAL ATTENUATION TABLE
C
C              S   THE SECOND DERIVATIVE TRANSFORM TABLE
C              S = EXP(-I * DR * L**2 / (2 * K)) / (NPTS / 2)
C              (RETURNED AS SR = REAL PART, SI = IMAGINARY PART)
C
C              U   THE INDEX OF REFRACTION TABLE
C              U = EXP(I * K * DR * (N**2 - 1) / 2)
C              (RETURNED AS UR = REAL PART, UI = IMAGINARY PART)
C
C              FACTOR = DR * K / 2
C
```

```
C                WHERE I = SQRT(-1)
C                      K = AVERAGE WAVE NUMBER
C                      N = INDEX OF REFRACTION
C
C     TEMPORARY VARIABLE - FL
C
C     SUBROUTINES - INDEX (CONSTRUCTS INDEX OF REFRACTION TABLE)
C
      SUBROUTINE SET
      COMMON /HERTZ/ CO,H,HK,F,FK,FACTOR,WL
C
      COMMON /MESH/ R , DR , NR , KR , DZ , ZMAX , IB , N , NPTS , N2 ,
     A              N4 , NL4 , NA , NW , ZW , HALF , NEFFW
C
      COMMON/FIL1/FIL(2047)
      COMMON /TABLE/ SR(2047),SI(2047),UR(2047),UI(2047),FN(2047)
C
C     GENERATE THE TABLE USED TO ATTENUATE THE ADDITIONAL DISCRETE
C     MODES INTRODUCED BY TRUNCATING THE FOURIER TRANSFORM.  THE
C     PHYSICAL BOTTOM IS EXTENDED 1/4 THE MAXIMUM DEPTH AND
C     TERMINATED WITH A HIGHLY ABSORBING LAYER.
C
C               2    2
C     GENERATE D E/DZ  TRANSFORM TABLE (INCLUDES FFT NORMALIZATION).
C
      NFIL = (3./4.) * FLOAT(NPTS+1)
      FACTOR=0.50*DR/FK
      DO 20 I=1,NPTS
      FL=H*FLOAT(I)
      FL=FACTOR*FL*FL
      SR(I)=COS(FL)/HALF
      SI(I)=-SIN(FL)/HALF
      IF(I.LE.NFIL) GO TO 20
      SR(I)=FIL(I)*SR(I)
      SI(I)=FIL(I)*SI(I)
   20 CONTINUE
C
C     GENERATE INDEX OF REFRACTION TABLE.
C
      FACTOR=0.50*DR*FK
      CALL INDEX
C
      RETURN
      END
C
C
C     INDEX GENERATES THE INDEX OF REFRACTION TABLE.
C
```

```
C     INPUT  - FACTOR = DR * K / 2
C              FN     = N**2 - 1
C              ATTEN  = VOLUME ATTENUATION
C
C     OUTPUT - U   INDEX OF REFRACTION TABLE
C              U=EXP(I*DR*(K*(N**2-1)/2+I*ATTEN))
C              (RETURNED AS UR = REAL PART, UI = IMAGINARY PART)
C
C              WHERE I = SQRT(-1)
C                    K = AVERAGE WAVE NUMBER
C                    N = INDEX OF REFRACTION
C
C     TEMPORARY VARIABLES - T, TR, TI
C
      SUBROUTINE INDEX
      COMMON /PHASE/ NC,Z(100),C(100),MC,DM(20)
      COMMON /HERTZ/ CO,H,HK,F,FK,FACTOR,WL
      COMMON /LOSFCN/ THETI(50),LOSSI(50),NLTH,ZHAT(100),ALGEN(100),
     1 HORRAN,JA,NHAT,RAPRES,KANEXT,CMUD(30),ZMUD(30),NMUD,NBOTM
      REAL LOSSI
C
      COMMON /MESH/ R , DR , NR , KR , DZ , ZMAX , IB , N , NPTS , N2 ,
     A              N4 , NL4 , NA , NW , ZW , HALF , NEFFW
C
      COMMON /TABLE/ SR(2047),SI(2047),UR(2047),UI(2047),FN(2047)
      COMMON /COSTR/ VABSF,ATTEN
      COMMON/FIL1/FIL(2047)
C
C
C        INTRODUCE VOLUME ABSORPTION/ATTENUATION
C
      IF(VABSF.GT.0.) ATTEN=-1.
      ALPHA=1.
      IF(ATTEN.GT.0.) ALPHA=EXP(-ATTEN*DR)
C
      NWC=NPTS
      IF(NBOTM.NE.0) NWC=NW
C
      DO 10 I=1,NWC
      T=FACTOR*FN(I)
      ABSLYR=FIL(I)
      UR(I)=ALPHA*COS(T)*ABSLYR
      UI(I)=ALPHA*SIN(T)*ABSLYR
   10 CONTINUE
C
      IF(NBOTM.EQ.0) RETURN
C
C        INTRODUCE ATTENUATING MUD SEGMENT TO PROFILE
```

```
C
      FN0=FN(NW)
      IF(NBOTM.LT.0) FN1=FNMUD(FN0,0)
      F1=(3.1415926535/HORRAN)
      ZM=0.0
C
      NWP1=NW+1
      DO 40 I=NWP1,NEFFW
      IF(NBOTM.LT.0) GO TO 20
C
C        USE ANALYTIC PROFILE IN MUD
C
      ZM=ZM+DZ
      CORR=ZM*F1
      ANMUD=FN0-CORR*CORR
      GO TO 30
C
C        USE USER-SPECIFIED PROFILE IN MUD - BUT FIRST SMOOTH PROFILE
C        WITH 1-2-1 FILTER TO TRY AND KEEP THE BUGGIES OUT
C
   20 FN2=FNMUD(FN1,1)
      ANMUD=0.25*(FN0+FN1+FN1+FN2)
      FN0=ANMUD
      FN1=FN2
C
C        BUILD INDEX TABLE WITH MUD
C
   30 T=FACTOR*ANMUD
      UR(I)=ALPHA*COS(T)*FIL(I)
      UI(I)=ALPHA*SIN(T)*FIL(I)
   40 CONTINUE
C
C        COMPLETE TABLE BY USING COSTANT SOUND SPEED GRADIENT
C
      NEFP1=NEFFW+1
      URC=UR(NEFFW)
      UIC=UI(NEFFW)
C
      DO 50 I=NEFP1,NPTS
      UR(I)=URC*FIL(I)
      UI(I)=UIC*FIL(I)
   50 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE FILLOS
C        INTRODUCE BOTTOM LOSS TO THE FILTER FUNCTION
```

```
C
      CCMMON /FIIT/ FIL(2047)
      CCMMON /MESH/ R,DR,NR,KR,DZ,ZMAX,IB,N,NPTS,N2,N4,NL4,NA,NW,ZW,
     1  HALF,NEFFW
      CCMMON /LOSFCN/ THETI(50),LOSSI(50),NLTH,ZHAT(100),ALGEN(100),
     1 HORRAN,JA,NHAT,RAPRES,RANEXT,CMUD(30),ZMUD(30),NMUD,NBOTM
      RFAL LCSSI
C
C        INITIALLY DETERMINE LOSS
C
      Z1 = 0.0
      K = 1
      NWP1 = NW + 1
      FIL(NW) = 0.0
      DC 30 I = NWP1,NEFFW
      Z1 = Z1 + DZ
   10 IF(K.GE.NHAT) GO TO 25
      IF( (Z1.GE.ZHAT(K)) .AND. (Z1.LE.ZHAT(K+1)))GC TO 20
      K = K + 1
      GO TO 10
C
   20 SIOPE = (AIGEN(K+1)-ALGEN(K))/(ZHAT(K+1)-ZHAT(K))
      AIZ = ALGEN(K) + SLOPE*(Z1-ZHAT(K))
      FIL(I) = AIZ*Z1
      GC TO 30
   25 FIL(I) = FII(I-1)
   30 CCNTINUE
C
C        MAKE SURE LOSS IS A RELATIVELY SMOOTH FUNCTICN OF DEPTH
C            BY APPLYING A 1-2-1 FILTER
C
      NEWM1 = NEFFW - 1
C
      DO 40 I = NWP1,NEWM1
      FIL(I) = 0.25*( FIL(I-1) + FIL(I) + FIL(I) + FIL(I+1))
   40 CONTINUE
C
C        CONVERT TO ATTENUATION IN PRESSURE
C
      DC 50 I = NW,NEFFW
      FIL(I) = EXP(-(2.302585/20.0)*FIL(I))
   50 CCNTINUE
C
      RETURN
      END
C
      FUNCTION FNMUD(FNF,K)
C
```

```
C          DETERMINE MODIFIED INDEX (N**2-1) IN BOTTOM ...
C               GIVEN USER-SPECIFIED PROFILE
C
      COMMON /MESH/ R,DR,NR,KR,DZ,ZMAX,IB,N,NPTS,N2,N4,NL4,NA,NW,ZW,
     1  HALF,NEFFW
      CCMMON /LOSFCN/ THETI(50),LOSSI(50),NLTH,ZHAT(100),ALGEN(100),
     1 HORRAN,JA,NHAT,RAPLES,RANEXT,CMUD(30),ZMUD(30),NMUD,NBOTM
      REAL LCSSI
      CCMMON /HERIZ/ CO,H,HK,F,FK,FACTOR,WL
      CCMMON /JUIK/ M,COLD,Z1
C
      IF(K.GT.0) GO TO 10
C
      M=1
      Z1=0.0
      CCLD = CO / SQRT(FNP+1.0)
C
   10 Z1=Z1+DZ
   20 IF((Z1.GE.ZMUD(M)).AND.Z1.LE.ZMUD(M+1))   GC TC 30
      CCLD=CCLD+(CMUD(M+1)-CMUD(M))
      M=M+1
      IF(M.LT.NMUD) GO TO 20
      CNEW=CCLD
      GO TO 40
C
   30 DELZ=(Z1-ZMUD(M))/(ZMUD(M+1)-ZMUD(M))
      CNEW = COLI + DELZ*(CMUD(M+1)-CMUD(M))
C
   4C FNN=CO/CNEW
      FNMUD=FNN*FNN-1.0
      RETURN
      END
C
C
C     FUNCTION TICSS INTERPOLATES THE FIELD AT DEPTH Z ANE
C     RETURNS THE TRANSMISSION LOSS.
C
C     INPUT   - RF   RECIPRCCAL RANGE
C               Z    DEPTH
C               DZ   MESH INCREMENT
C               PR   REAI PART OF FIELD MESH
C               PI   IMAGINARY PART OF FIELD MESH
C
C     CUTPUT - TICSS TRANSMISSION LOSS AT DEPTH Z
      FUNCTION TICSS(RR,Z)
C
      CCMMON /FIEID/ PR(2049),PI(2049)
C
```

```
      COMMON /MESH/ R , DR , NR , KR , DZ , ZMAX , IB , N , NPTS , N2 ,
     A              N4 , NI4 , NA , NW , ZW , HALF , NEFFW
C
      DATA PMIN/1.0E-18/
C
      ZM=Z/DZ
      M=ZM
      PZ=0.
      IF (M.GT.0) PZ=SQRT(PR(M)*PR(M)+PI(M)*PI(M))
      PZ=PZ+(SQRT(PR(M+1)*PR(M+1)+PI(M+1)*PI(M+1))-PZ)*(ZM-FLOAT(M))
      TLOSS=-10.*ALOG10(RR*PZ*PZ+PMIN)
      RETURN
      END
      FUNCTION SPEED(D)
C
C     SPEED RETURNS THE SOUND SPEED AT DEPTH = D.
C
C     INPUT  - NC  NUMBER OF POINTS IN SOUND VELOCITY PROFILE TABLE.
C               Z  DEPTH ARRAY.
C               C  SOUND SPEED ARRAY.
C               D  INPUT DEPTH.
C
C     OUTPUT - SPEED  SOUND SPEED AT DEPTH D.
C
      COMMON /PHASE/ NC,Z(100),C(100),MC,DM(20)
C
      K=2
C
10    IF (D.LT.Z(K)) GO TO 20
      K=K+1
      IF (K.LT.NC) GO TO 10
20    SPEED=C(K-1)+(C(K)-C(K-1))*(D-Z(K-1))/(Z(K)-Z(K-1))
C
      RETURN
      END
C
C
C     FILTER EVALUATES THE INDEX OF REFRACTION ON THE FIELD MESH AND
C     TRANSFORMS THE ENVIRONMENT TO REDUCE THE PARABOLIC PHASE
C     VELOCITY ERROR.
C
C     INPUT  - NC  NUMBER OF POINTS ON SOUND VELOCITY PROFILE
C               Z  DEPTH ARRAY (NC DEPTHS)
C               C  SOUND SPEED ARRAY (NC SOUND SPEEDS)
C              C0  REFERENCE SOUND SPEED
C              MC  PHASE VELOCITY CORRECTION FLAG
C               D  OUTPUT DEPTH ARRAY
C
```

```
C       OUTPUT - FN   SMOOTHED INDEX OF REFRACTION ARRAY
C                DM  TRANSFORMED OUTPUT DEPTH ARRAY
C                CD  TRANSFORMED FIELD PLOT DEPTH ARRAY
C
C       LOCAL VARIABLES - ZM   DEPTH (FT) AT FIELD MESH POINT
C                         CM   INDEX OF REFRACTION AT FIELD MESH POINT
C                         G    SOUND SPEED GRADIENT
C
C       CONSTANTS - FT  CONVERSION FACTOR METERS/FT
C
        SUBROUTINE FILTER(ND,D)
        DIMENSION D(20)
        COMMON /UNITS/ LC,LP,IT
        COMMON /HERTZ/ CO,H,HK,F,FK,FACTOR,WL
        COMMON /PHASE/ NC,Z(100),C(100),MC,DM(20)
        COMMON/PLT/TITLE(20),NPLT,LCR,CLMIN,DCL,CD1,CD2,DCD,CD(120)
C
        COMMON /MESH/ R,DR,NR,KR,DZ,ZMAX,IB,N,NPTS,N2,N4,NL4,NA,NW,ZW,
       1  HALF,NEFFK
C
        COMMON /TABLE/ SR(2047),SI(2047),UR(2047),UI(2047),FN(2047)
        DATA FT/0.3048/
C
        IF (Z(1).EQ.0.) GO TO 10
C
        WRITE(LP,900)
900     FORMAT(32H0INVALID SOUND VELOCITY PROFILE./
       1       40H SOUND SPEED NOT DEFINED AT THE SURFACE.)
        STOP
C
C       CHECK INPUT PROFILE UNITS.
C
10      IF (C(1).GT.3000.) GO TO 30
C
C       CONVERT UNITS TO FT AND FT/SEC.
C
        DO 20 I=1,NC
        Z(I)=Z(I)/FT
20      C(I)=C(I)/FT
C
30      IF (NC.GE.2) GO TO 40
C
        NC=2
        Z(2)=DZ*FLOAT(NW)
        C(2)=C(1)
C
C       INTERPOLATE N**2 - 1 ON THE FIELD MESH.
C
```

```
40     K=0
       L=NC-1
       ZP=0.
       ZW=DZ*FLOAT(NW)
       IF (MC.EQ.1) ZW=ZW*SQRT(C0/SPEED(ZW))
       ASSIGN 45 TO LOOP
C
       I=1
45     ZM=DZ*FLOAT(I)
       IF (ZM.LE.ZW) GO TO 50
       ASSIGN 70 TO LOOP
       GO TO 70
50     IF (ZM.LT.ZP.OR.K.EQ.L) GO TO (1,2),MC
       K=K+1
       G=(C(K+1)-C(K))/(Z(K+1)-Z(K))
       ZP=Z(K+1)
       IF (MC.EQ.1) ZP=ZP*SQRT(C0/C(K+1))
       GO TO 50
1      CM=G*ZM*ZM
       ZM=(CM+ZM*SQRT(G*CM+4.*C0*(C(K)-G*Z(K))))/(C0+C0)
2      CM=C0/(C(K)+G*(ZM-Z(K)))
       IF (MC.EQ.1) GO TO 60
       CM=CM*CM-1.0
       GO TO 70
60     CM=CM+CM-2.0
70     FN(I)=CM
       I=I+1
       IF (I.LE.NPTS) GO TO LOOP,(45,70)
C
C      SMOOTH THE MESH WITH A 1-2-1 FILTER.
C
       L=NPTS-1
       FN(1) = 0.25*(FN(1) + FN(1) + FN(1) + FN(2))
       DO 80 I=2,L
80     FN(I)=0.25*(FN(I-1)+FN(I)+FN(I)+FN(I+1))
C
       IF (MC.EQ.2) GO TO 100
C
C      TRANSFORM THE OUTPUT AND FIELD PLOT DEPTHS.
C
       DO 90 I=1,ND
90     DM(I)=D(I)*SQRT(C0/SPEED(D(I)))
C
       IF (NPLT.LE.0) GO TO 100
C
       ZM=CD1
       DO 95 I=1,NPLT
       CD(I)=ZM*SQRT(C0/SPEED(ZM))
```

```
95     ZM=ZM+DCD
C
100    RETURN
       END
C
C
C
C      SOURCE CONSTRUCTS THE INITIAL FIELD.
C      THE FIELD IS DEFINED AS A GAUSSIAN BEAM AT RANGE = 0.
C
C      INPUT   - ZS  INPUT DEPTH (FT)
C                FK  AVERAGE WAVE NUMBER
C                DZ  DEPTH MESH INCREMENT  (FT)
C
C      OUTPUT - PR  REAL PART OF FIELD
C                PI  IMAGINARY PART OF FIELD
C
C      LOCAL VARIABLES - GA  GAUSSIAN AMPLITUDE
C                        GW  GAUSSIAN WIDTH
C                        ZM  DEPTH (FT) OF FIELD MESH POINT
C
       SUBROUTINE SOURCE(ZS)
       COMMON/FILT/FIL(2047)
       COMMON /FIELD/ PR(2049),PI(2049)
       COMMON /HERTZ/ CO,H,HK,F,FK,FACTOR,WL
C
C      CODE BETWEEN $$$$$$$ HAS BEEN ADDED TO ENABLE WAVENUMBER
C      TECHNIQUE FOR NPS INSTALLATION
C
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
       COMMON /WTFLD/WPR(2049),WPI(2049),PRS(2049),GA
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C
       COMMON /MESH/ R , DR , NR , KR , DZ , ZMAX , IB , N , NPTS , N2 ,
     1             N4 , NL4 , NA , NW , ZW , HALF , NEFFW
C
       DIMENSION HI(33)
       DATA HI/.82238912,   .16802293,  -.14123685,  .10267275,
     $   -.059805428, .020310007,  .9728278E-02,  -.27054637E-01,
     $   .031638565,  -.26196492E-01,  .15072034E-01, -.28659978E-02,
     $  -.67669929E-02, .11843987E-01,  -.12178481E-01, .89802489E-02,
     $  -.41511850E-02, -.45092861E-03,  .35526239E-02, -.46902440E-02,
     $  .41261945E-02,  -.25613065E-02,  .78673982E-03,  .5931843E-03,
     $ -.12965481E-02, .13517565E-02, -.93507139E-03,  .47731731E-03,
     $ -.49001725E-04,-.19387227E-03,.25337986E-03,-.19606612E-03,
     $ .11718947E-03/
       DATA NFIR/33/
C
```

```
C
C                     INITIAIIZE ARRAYS
C
C
C
      NM2=NPTS+2
      DC 15 I=1,NM2
      PR(I) = 0.0
      PI(I) = 0.0
   15 CCNTINUE
C
C                     INTRODUCE FILTER IMPULSE RESPCNSE
C
      DC 20 I=1,NFIR
      PR(I)=HI(I)
   20 CONTINUE
C
C                     TRANSFORM TO VERTICAL WAVENUMEER SPACE
C
C
      CALL RST(PR,N,0)
C
C                     FINALLY GET A CHANCE TO INTRODCUCE SOURCE
C
C
C
C
C
C
C                     FILL THE FILTER WITH ONE
C
      NFIL1 = (3./4.)*FLOAT(NPTS+1)
C
      GA=2.*HALF*SQRT(WI)/ZMAX
C
      DARG=H*ZS
      ARG=DARG
      DC 30 I=1,NPTS
      FIL(I)=1.
      IF (I.GT.NFIL1)  FIL(I)=2.*PR(I)
      PR(I)=2.*PR(I)*GA*SIN(ARG)
C
C     CCDE BETWEEN $$$$$$$ HAS BEEN ADDED TO ENABLE WAVENUMBER
C     TECHNIQUE FCR NPS INSTALLATION
C
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
      PRS(I) = PR(I)
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
      ARG=ARG+DARG
```

```
30    CONTINUE
C
C
      RETURN
      END
C
C
C     FLD IS A LINE PRINTER FIELD PLOTTING ROUTINE.  SYMBOLS
C     CORRESPONDING TO ICL DB TRANSMISSION LOSS BINS AT UP
C     TO 120 DEPTH MESH POINTS FROM DEPTH CD1 TO DEPTH CD2 ARE PRINTED
C     AT APPROXIMATELY A 1 NAUTICAL MILE RANGE INCREMENT.  A DEPTH
C     SCALE AND A TRANSMISSION LOSS SCALE ARE PRINTED THE FIRST TIME
C     FLD IS CALLED.
C
C     INPUT - NPLT  NUMBER OF PLOT DEPTHS (.LE. 120)
C             CD1   MINIMUM PLOT DEPTH
C             DCD   PLOT DEPTH INCREMENT
C             CLMIN MINIMUM LOSS
C             DCL   LOSS INCREMENT
C             CD    PLOT DEPTH ARRAY
C             ZW    WATER DEPTH
C             R     CURRENT RANGE
C             DR    CURRENT RANGE STEP
C             RR    RECIPROCAL RANGE
C
C     LOCAL VARIABLES - KCR    CURRENT RANGE
C                       NUM    SYMBOLS FOR PLOTTING DEPTHS
C                       LEVEL  PLOTTING SYMBOL ARRAY
C                       PL     TRANSMISSION LOSS AT DEPTH CD(I)
C                       CL     CURRENT PLOT LOSS BIN
C
C     SUBROUTINE - FUNCTION TLOSS (INTERPOLATE FIELD AND RETURN
C                                  TRANSMISSION LOSS)
C
C
      SUBROUTINE FLD(RR)
      INTEGER TITLE
      DIMENSION LEVEL(6),NUM(10),LOSS(120)
      COMMON /FIELD/ PR(2049),PI(2049)
      COMMON /UNITS/ LC,LP,IT
      COMMON /OUTPUT/ NOUT,RNM,TL(20)
      COMMON/PLT/TITLE(20),NPLT,LCR,CLMIN,DCL,CD1,CD2,DCD,CD(120)
C
      COMMON /MESH/ R , DR , NR , KR , DZ , ZMAX , IB , N , NPTS , N2 ,
     A              N4 , NL4 , NA , NW , ZW , HALF , NEFFW
C
      DATA FNM,LEVEL/6076.1,1H+,1H/,1H ,1H*,1H0,1HB/
      DATA NUM/1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9/
```

```
C
      KCR=(R+0.50*DR)/FNM
      IF (KCR.EQ.ICR) RETURN
C
C     CHECK FOR FIRST CALL.
C
      IF (LCR.GT.0) GO TO 40
C
      WRITE(LP,900) TITLE
  900 FORMAT(1H1,20A4)
C
C     PRINT TRANSMISSION LOSS SCALE.
C
      K=4
C
      DO 2 I=1,5
      DO 1 J=1,10
      K=K+1
1     LOSS(K)=LEVEL(I)
2     CONTINUE
C
      CL=CLMIN
C
      DO 3 I=1,4
      LOSS(I)=CL+0.50
      IF (LOSS(I).LT.100) LOSS(10*I+5)=LEVEL(I)
3     CL=CL+DCL
C
      WRITE(LP,910) (LOSS(I),I=1,K)
910   FORMAT(1H0,21X,8HTL SCALE/11X,4(I3,3H DB,4X)/1X,50A1)
C
C     PRINT DEPTH SCALE.
C
      WRITE(LP,915)
915   FORMAT(1H0,62X,10HDEPTH (FT))
C
      L=10000
C
      DO 30 I=1,5
      K=NPLT+1
      DO 20 J=1,NPLT
      K=K-1
      IZ=CD1+DCD*FLOAT(J-1)
      IF (L.EQ.1.OR.IZ.GE.L) GO TO 10
      LOSS(K)=LEVEL(3)
      GO TO 20
10    KZ=1+IZ/L-10*(IZ/(10*L))
      LOSS(K)=NUM(KZ)
```

```
20      CONTINUE
        L=L/10
30      WRITE(LP,920) (LOSS(K),K=1,NPLT)
920     FORMAT(9X,120A1)
C
        WRITE(LP,930)
930     FORMAT(2X,5HRANGE,2X,120(1H-))
C
C       PRINT TRANSMISSION LOSS SYMBOLS AT PLOT DEPTHS.
C
40      LCR=KCR
        K=NPLT+1
C
C
C                         CHANGE INTENSITY TO PRESSURE
C
        CLMPRS=10.**(-CLMIN/10.)
        DCLPRS=10.**(-DCL/10.)
C
C
C
        DO 60  I=1,NPLT
        K=K-1
        IF (CD(I).LT.ZW)  GO TO 50
        L=6
        GO TO 60
50      L=1
        CL=CLMPRS
C
C
C                         CALCULATING THE PRESSURE BY AVERAGING
C
        ZM=CD(I)/DZ
        M=ZM
        PL=0.
        IF(M.GT.0)  PL=PR(M)**2+PI(M)**2
        PL=PL+(PR(M+1)**2+PI(M+1)**2-PL)*(ZM-FLOAT(M))**2
        PL=RR*PL
C
C
55      IF (PL.GT.CL)  GO TO 60
        CL=CL*DCLPRS
        L=L+1
        IF (L.LT.5) GO TO 55
60      LOSS(K)=LEVEL(L)
C
        WRITE(LP,940) RNM, (LOSS(K),K=1,NPLT)
940     FORMAT(1X,F7.2,1X,120A1)
```

```
C
      RETURN
      END
C
C
C     RST REPLACES THE REAL VECTOR X BY ITS FINITE DISCRETE SINE
C     TRANSFORM.  THE ALGORITHM IS BASED ON A MIXED RADIX (8-4-2)
C     REAL VECTOR FAST FOURIER SYNTHESIS ROUTINE PUBLISHED BY BERGLAND
C     (G. D. BERGLAND, "A RADIX-EIGHT FAST FOURIER TRANSFORM SUBROUTINE
C     FOR REAL-VALUED SERIES," IEEE TRANSACTIONS ON AUDIO AND ELECTRO-
C     ACOUSTICS, VOL. AU-17, PP. 138-144, JUNE 1969) AND A SINE
C     TRANSFORM ALGORITHM PUBLISHED BY COOLEY, LEWIS, AND WELSH
C     (J. W. COOLEY, P. A. W. LEWIS AND P. D. WELSH, "THE FAST FOURIER
C     TRANSFORM ALGORITHM PROGRAMMING CONSIDERATIONS IN THE CALCULATION
C     OF SINE, COSINE AND LAPLACE TRANSFORMS," J. SOUND VIB., VOL. 12,
C     PP. 315-337, JULY 1970).
C
C     INPUT  - X  A 2**N - 1 POINT REAL ARRAY
C              N  TRANSFORM SIZE
C     OUTPUT - X
C                  FIRST TIME THRU COSINE TRANSFORM OF INPUT VECTOR X
C                  ALL OTHER TIMES SINE TRANSFORM
C
C              IFLAG =0  DO COSINE TRANSFORM
C                     1  DO A  SINE TRANSFORM
C
C     TABLES - ARRAY   REQUIRED DIMENSIONS
C               E         2**N
C               ST        2**N - 1
C               JI        2**(N-1) - 1
C               CS        2**(N-4) - 1
C               SS        2**(N-4) - 1
C
C     SUBROUTINES - R8SYN (RADIX 8 SYNTHESIS)
C                   R4SYN (RADIX 4 SYNTHESIS)
C                   R2TR  (RADIX 2 TRANSFORM)
C
C
      SUBROUTINE RST(X,N,IFLAG)
      DIMENSION X(2),B(2049),ST(2049),JI(1024)
      COMMON /WTS/ N1,CS(128),SS(128)
      DATA N2,PI/0,3.14159265358979/
C
      IF (N.NE.N2) GO TO 200
C
C
10    CONTINUE
      IF (IFLAG.GT.0) GO TO 30
```

```
C
C     SET UP THE ARRAY FOR THE COSINE TRANSFORM
C
C
      B(1)=X(1)
      B(2)=X(NP+1)
      J1=0
      DO 20 J=3,NPM1,2
      J1=J1+1
      J2=JI(J1)
      J3=NP-J2
      B(J)=X(J2+1)
 20   B(J+1)=X(J3+2)-X(J3)
      GO TO 45
C
C     SET UP ARRAY FOR THE SINE TRANSFORM
C
 30   CONTINUE
      B(1)=-X(1)-X(1)
      B(2)=X(NPM1)+X(NPM1)
C
      J1=0
      DO 40 J=3,NPM1,2
      J1=J1+1
      J2=JI(J1)
      J3=NP-J2
      B(J)=X(J2-1)-X(J2+1)
 40   B(J+1)=X(J3)
 45   CONTINUE
C
C     BEGIN FAST FOURIER SYNTHESIS
C
      IF (N8.EQ.0) GO TO 60
C
C     RADIX 8 ITERATIONS
C
      INT=1
      NT=NPD16
      DO 50 J=1,N8
      J1= 1+INT
      J2=J1+INT
      J3=J2+INT
      J4=J3+INT
      J5=J4+INT
      J6=J5+INT
      J7=J6+INT
      CALL R8SYN(INT,B,B(J1),B(J2),B(J3),B(J4),B(J5),B(J6),B(J7))
      NT=NT/8
```

```
      INT=8*INT
50    CONTINUE
C
60    IF (N4) 90,80,70
C
C     RADIX 4 ITERATION
C
70    INT=NPD4
      J1= 1+INT
      J2=J1+INT
      J3=J2+INT
      CALL R4SYN(INT,B,B(J1),B(J2),B(J3))
C
      GO TO 90
C
C     RADIX 2 ITERATION
C
80    INT=NPD2
      J1= 1+INT
      CALL R2TR(INT,B,B(J1))
C
C
90    CONTINUE
      J1=NP
      IF (IFLAG.GT.0) GO TO 95
C
C     FORM THE COSINE TRANSFORM
C
      DO 94 J=1,NPM1
      X(J)=.25*((B(J1)-B(J+1))*ST(J)+B(J+1)+B(J1))
      J1=J1-1
94    CONTINUE
      RETURN
C
C     FORM SINE TRANSFORM
C
95    CONTINUE
      DO 100 J=1,NPM1
      X(J)=0.25*((B(J+1)+B(J1))*ST(J)-B(J+1)+B(J1))
100   J1=J1-1
C
      RETURN
C
C     COMPUTE CONSTANTS AND CONSTRUCT TABLES
C
200   N2=N
      N8=N2/3
      N4=N2-3*N8-1
```

```
      NP=2**N2
      NPD2=NP/2
      NPD4=NP/4
      NPD16=NP/16
      NPM1=NP-1
      DT=PI/FLOAT(NP)
C
      DO 210 J=1,NPM1
      T=DT*FLOAT(J)
210   ST(J)=0.50/SIN(T)
C
C     CONSTRUCT THE BIT REVERSED SUBSCRIPT TABIE.
C
      J1=0
      NT=NPD2-1
      DC 240 J=1,NT
      J2=NPD2
220   IF (AND(J1,J2).EQ.0) GO TO 230
      J1=IABS(J1-J2)
      J2=J2/2
      GC TO 220
230   J1=J1+J2
240   JI(J)=J1
C
      IF (N8.EQ.C) GO TO 10
C
C     CONSTRUCT THE TRIGONCMETRIC TABLES FOR THE RADIX 8 PASSES.
C     THE TABLES ARE STORED IN BIT REVERSED ORDER.
C
      J1=0
      NT=NPD16-1
      DC 270 J=1,NT
      J2=NPD16
250   IF (AND(J1,J2).EQ.0) GO TO 260
      J1=IABS(J1-J2)
      J2=J2/2
      GC TO 250
260   J1=J1+J2
      T=DT*FLOAT(J1)
      CS(J)=COS(T)
270   SS(J)=-SIN(T)
C
      GO TO 10
C
      END
C
      SUBROUTINE R8SYN(INT,B0,B1,B2,B3,B4,B5,B6,B7)
C
```

```
C     RADIX 8 SYNTHESIS SUBROUTINE
C     CALLED BY RST, THE SINE TRANSFORM DRIVER.
C
      DIMENSION B0(2),B1(2),B2(2),B3(2),B4(2),B5(2),B6(2),B7(2)
      CCMMON /WTS/ NT,CS(128),SS(128)
      DATA R2,CPI4/1.41421356237310,0.70710678118655/
      DATA CPI8,SPI8/0.92387953251129,0.38268343236509/
C
      JT=0
      JI=2
      JR=2
      JI=3
      INT8=8*INT
C
      DC 72 K=1,INT
      T0=B0(K)+B1(K)
      T1=B0(K)-B1(K)
      T2=B2(K)+B2(K)
      T3=B3(K)+B3(K)
      T4=B4(K)+B6(K)
      T5=B4(K)-B6(K)
      T6=B7(K)-B5(K)
      T7=B7(K)+B5(K)
      T8=R2*(T7-T5)
      T5=R2*(T7+T5)
      TT0=T0+T2
      T2=T0-T2
      TT1=T1+T3
      T3=T1-T3
      T4=T4+T4
      T6=T6+T6
      B0(K)=TT0+T4
      B4(K)=TT0-T4
      B1(K)=TT1+T5
      B5(K)=TT1-T5
      B2(K)=T2+T6
      B6(K)=T2-T6
      B3(K)=T3+T8
      B7(K)=T3-T8
   72 CCNTINUE
C
      IF (NT.EQ.0) GO TC 70
C
      K0=INT8+1
      KLAST=K0+INT-1
C
      DO 75 K=K0,KLAST
      T1=B0(K)+B6(K)
```

```
      T3=B0(K)-B6(K)
      T2=B7(K)-B1(K)
      T4=B7(K)+B1(K)
      T5=B2(K)+B4(K)
      T7=B2(K)-B4(K)
      T6=B5(K)-B3(K)
      T8=B5(K)+B3(K)
      B0(K)=(T1+T5)+(T1+T5)
      B4(K)=(T2+T6)+(T2+T6)
      T5=T1-T5
      T6=T2-T6
      B2(K)=R2*(T6+T5)
      B6(K)=R2*(T6-T5)
      T1=T3*CPI8+T4*SPI8
      T2=T4*CPI8-T3*SPI8
      T3=T8*CPI8-T7*SPI8
      T4=-T7*CPI8-T8*SPI8
      B1(K)=(T1+T3)+(T1+T3)
      B5(K)=(T2+T4)+(T2+T4)
      T3=T1-T3
      T4=T2-T4
      B3(K)=R2*(T4+T3)
      B7(K)=R2*(T4-T3)
   75 CONTINUE
C
      GO TO 70
C
 76   C1=CS(JT)
      S1=SS(JT)
      C2=C1*C1-S1*S1
      S2=C1*S1+C1*S1
      C3=C1*C2-S1*S2
      S3=C2*S1+S2*C1
      C4=C2*C2-S2*S2
      S4=C2*S2+C2*S2
      C5=C2*C3-S2*S3
      S5=C3*S2+S3*C2
      C6=C3*C3-S3*S3
      S6=C3*S3+C3*S3
      C7=C3*C4-S3*S4
      S7=C4*S3+S4*C3
C
      K=JI*INT8
      J0=JR*INT8+1
      JLAST=J0+INT-1
C
      DO 77  J=J0,JLAST
      K=K+1
```

```
TR0=B0(J)+B6(K)
TR1=B0(J)-B6(K)
TI0=B7(K)-B1(J)
TI1=B7(K)+B1(J)
TR2=B4(K)+B2(J)
TI3=B4(K)-B2(J)
TI2=B5(K)-B3(J)
TR3=B5(K)+B3(J)
TR4=B4(J)+B2(K)
T0=B4(J)-B2(K)
TI4=B3(K)-B5(J)
T1=B3(K)+B5(J)
TR5=CPI4*(T1+T0)
TI5=CPI4*(T1-T0)
TR6=B6(J)+B0(K)
T0=B6(J)-B0(K)
TI6=B1(K)-B7(J)
T1=B1(K)+B7(J)
TR7=-CPI4*(T0-T1)
TI7=-CPI4*(T0+T1)
T0=TR0+TR2
TR2=TR0-TR2
T1=TI0+TI2
TI2=TI0-TI2
T2=TR1+TR3
TR3=TR1-TR3
T3=TI1+TI3
TI3=TI1-TI3
T5 =TI4+TI6
TTR6=TI4-TI6
TI6=TR6-TR4
T4 =TR4+TR6
T7 =TI5+TI7
TTR7=TI5-TI7
TI7=TR7-TR5
T6 =TR5+TR7
B0(J)=T0+T4
B0(K)=T1+T5
B1(J)=C1*(T2+T6)-S1*(T3+T7)
B1(K)=C1*(T3+T7)+S1*(T2+T6)
B2(J)=C2*(TR2+TTR6)-S2*(TI2+TI6)
B2(K)=C2*(TI2+TI6)+S2*(TR2+TTR6)
B3(J)=C3*(TR3+TTR7)-S3*(TI3+TI7)
B3(K)=C3*(TI3+TI7)+S3*(TR3+TTR7)
B4(J)=C4*(T0-T4)-S4*(T1-T5)
B4(K)=C4*(T1-T5)+S4*(T0-T4)
B5(J)=C5*(T2-T6)-S5*(T3-T7)
B5(K)=C5*(T3-T7)+S5*(T2-T6)
```

```
      B6(J) =C6*(TR2-TTR6)-S6*(TI2-TI6)
      B6(K) =C6*(TI2-TI6)+S6*(TR2-TTR6)
      B7(J) =C7*(TR3-TTR7)-S7*(TI3-TI7)
      B7(K) =C7*(TI3-TI7)+S7*(TR3-TTR7)
   77 CONTINUE
C
      JR=JR+2
      JI=JI-2
      IF (JI.GT.JI) GO TO 70
      JI=JR+JR-1
      JI=JR
70    JT=JT+1
      IF (JT.LT.NT) GO TO 76
C
      RETURN
      END
C
      SUBROUTINE R4SYN(INT,B0,B1,B2,B3)
C
C     RADIX 4 SYNTHESIS SUBROUTINE
C     CALLED BY RST, THE SINE TRANSFORM DRIVER.
C
      DIMENSION B0(2),B1(2),B2(2),B3(2)
C
      DO 200 K=1,INT
      T0=B0(K)+B1(K)
      T1=B0(K)-B1(K)
      T2=B2(K)+B2(K)
      T3=B3(K)+B3(K)
      B0(K)=T0+T2
      B2(K)=T0-T2
      B1(K)=T1+T3
      B3(K)=T1-T3
200   CONTINUE
C
      RETURN
      END
C
      SUBROUTINE R2TR(INT,B0,B1)
C
C     RADIX 2 TRANSFORM SUBROUTINE
C     CALLED BY RST, THE SINE TRANSFORM DRIVER.
C
      DIMENSION B0(2),B1(2)
C
      DO 100 K=1,INT
      T=B0(K)+B1(K)
      B1(K)=B0(K)-B1(K)
```

```
      BO(K)=T
100   CONTINUE
C
      RETURN
      END
C
      FUNCTION ZB(R)
C
C     ZB RETURNS THE BOTTOM DEPTH AT RANGE R.
C
      COMMON /BATHY/ RB,KB,NB,BR(101),BZ(101)
C
10    IF (R.LT.BR(KB+1)) GO TO 20
      KB=KB+1
      BS=(BZ(KB+1)-BZ(KB))/(BR(KB+1)-BR(KB))
      GO TO 10
C
20    ZB=BZ(KB)+BS*(R-BR(KB))
C
      RETURN
      END
C
      SUBROUTINE SVP(NC,Z,C,RNEXT)
C
      COMMON/UNITS/LC,LP,LT
C
      DIMENSION Z(2),C(2)
C
      READ(2)NC,(Z(I),C(I),I=1,NC)
      READ(2)RNEXT
C
      RETURN
      END
C
C
C*************************************************************************
C
C         THIS ROUTINE GENERATES THE ATTENUATION FUNCTION ALPHA(Z)
C         ON A MESH ZHAT GIVEN THE FOLLOWING
C              A TABULATED L(THETA) CURVE AS GIVEN IN LOSSI(),THETI()
C              THE HORIZONTAL RANGE PERIOD HORRAN
C              A SOUND SPEED PROFILE OF INVERSE PARABOLIC FORM DERIVED
C                   SUCH THAT THE RAY PERIOD IN THE MEDIUM IS
C                   INDEPENDANT OF RAY GRAZING ANGLE.
C
C              THE FUNCTION ALPHA IS CONSTRUCTED AT MESH POINTS GIVEN
C                   BY ZHAT().. ALPHA IS REPRESENTED BY A PIECEWISE
C                   CONTINUOUS FUNCTION IN DEPTH, WITH THE COEFFICIENTS
```

```
C                 IN EACH LINEAR SEGMENT GIVEN IN CI() AND DI().
C
C***********************************************************************
C
      SUBROUTINE IOSGEN
      CCMMON /LOSFCN/ THETI(50),LOSSI(50),NLTH,ZHAT(100),ALGEN(100),
     A HORRAN,JA,NHAT,RAPRES,KANEXT,CMUD(30),ZMUD(30),NMUD,NBCTM
      REAL LCSSI
C
      DIMENSION T1(100),T2(100),CI(100),DI(100)
      REAL LOFZ,INTEG1,INTEG2
      CCMMON / HERTZ / CO,H,HK,F,FK,FACTOR,WL
      CCMMON /UNITS/ LC,LP,IT
      CCNA = HORRAN / 3.1415926
      GMAX = -10CC.0
      IF(LOSSI(1).GT.0.0) GMAX=AMAX1(GMAX,LOSSI(1)/(180.0*HK/3.1415926))
      DO 5 J = 2 , NLTH
      GRAD = (LOSSI(J)-LOSSI(J-1)) / (THETI(J)-THETI(J-1))
      GMAX = AMAX1(GMAX,GRAD)
    5 CCNTINUE
      IF(GMAX .GT. 1.0) WRITE(LP,6)
      IF(GMAX .GT. 1.0) STCP
    6 FCRMAT(80HOMAXIMUM LOSS GRADIENT HAS EXCEEDED 1.0 DB/DEGREE. PE CO
     XDE IS UNABIE TO HANDIE.         )
C             MAXIMUM ZHAT AND INCREMENT
      NHAT = 1
      ZHAT(1) = C.0
      ZINC = 250.0
   10 CONTINUE
      NHAT = NHAT + 1
      ZHAT(NHAT) = ZHAT(NHAT-1) + ZINC
      IF(ZHAT(NHAT) .LT. (HORRAN/5.0)   ) GO TO 10
C        INITIALIZE THE ATTENUATION COEFFICIENTS AND PARTIAL INTEGRALS
      DC 12 I=1 , 100
      CI(I) = 0.0
      DI(I) = 0.C
      T1(I) = 0.0
      T2(I) = 0.C
   12 CCNTINUE
C        INSERT ZHAT POINTS CORRESPONDING TO TABULATED L(THETA)
      DC 130 I = 1 , NLTH
      THET=THETI(I)
      ZTHI = HORRAN*SIN(3.1415926*THET/180.) / 3.1415926
      NM1 = NHAT-1
      DC 110 J = 1 , NM1
      IF(ZTHI .GT. ZHAT(J) .AND. ZTHI .LT. ZHAT(J+1)) GO TO 105
      GO TO 110
  105 IMOVE = NHAT - (J+1) + 1
```

```
      DO 106 L = 1 , LMOVE
      M = NHAT + 2 - L
      ZHAT(M)=ZHAT(M-1)
  106 CONTINUE
      ZHAT(M-1)  = ZTHI
      NHAT = NHAT + 1
      GO TO 130
  110 CONTINUE
  130 CONTINUE
C        START LOOP OVER TABULATED POINTS IN FINAL MESH
      J=2
  300 CONTINUE
      IFLAG = 0
  301 CONTINUE
      ZHJ =  ZHAT(J)
      ALOSZJ = LOFZ(THETI,LOSSI,NLTH,ZHJ,HORRAN)    - LOSSI(1)
C        START LOOP ON PARTIAL INTEGRALS
      JM1 = J - 1
      DO 20 K = 1 , JM1
      T1(K+1)  =  INTEG1(ZHAT(K),ZHAT(K+1),ZHJ,CONA)
      T2(K+1)  =  INTEG2(ZHAT(K),ZHAT(K+1),ZHJ,CONA)
   20 CONTINUE
      ALHS = ALOSZJ
      IF(J .LE. 2) GO TO 26
C        SUM OVER ALL SEGMENTS BUT THE LAST ONE
      LIMU = J - 1
      DO 24 I = 2 , LIMU
      ALHS = ALHS - DI(I)*T1(I)  - CI(I)*T2(I)
   24 CONTINUE
      ALHS = ALHS -  (CI(J-1)  + DI(J-1)*ZHAT(J-1))*T2(J)
   26 CONTINUE
      DI(J)  = ALHS / (T1(J)  - ZHAT(J-1)*T2(J))
      CI(J)  = CI(J-1)  +  (DI(J-1)-DI(J))*ZHAT(J-1)
C        COMPUTE ALPHA ON INTERVAL BOUNDING POINTS
      BLO = DI(J)*ZHAT(J-1)+CI(J)
      BHI=DI(J)*ZHAT(J)+CI(J)
      IF(BLO .GE.0.0  .AND. BHI .GE. 0.0) GO TO 303
C
C        ALPHA WENT NEGATIVE ON THIS INTERVAL ... INSERT A POINT INTO
C        ZHAT BY BISECTING THE CURRENT INTERVAL. IFLAG IS A BISECTION
C        COUNTER FOR THIS INTERVAL
      ZHN=(ZHAT(J)+ZHAT(J-1)) / 2.0
      IF(IFLAG .GT. 0)  GO TO 304
      NHAT = NHAT + 1
      K = NHAT
      NMOVE = NHAT - J
      DO 306 M = 1 , NMOVE
      ZHAT(K) = ZHAT(K-1)
```

```
      K = K - 1
  306 CONTINUE
  304 CONTINUE
      ZHAT(J) = ZHN
  305 IFLAG = IFLAG + 1
C        CHECK FOR EXCESSIVE BISECTION
      IF(IFLAG.LT.6) GO TO 301
      WRITE(LP,307)
      STOP
  307 FORMAT(85HCBOTTOM ATTENUATION CALCULATION TERMINATED ... UNABLE TO
     X REPRESENT ALPHA AS LINEAR                        )
  303 CONTINUE
      IFLAG = 0
      J = J + 1
      IF(J.LE.NHAT) GO TO 300
C
C        COMPUTE ALPHA(Z) AT THE TABULATED POINTS ZHAT()
C        JA WILL BE INDEX OF MAXIMUM ALPHA
      ALGEN(1) = 0.0
      DO 310 J = 2 , NHAT
      ALGEN(J) = 0.5 * (CI(J) + DI(J)*ZHAT(J) )
  310 CONTINUE
      IF(LOSSI(1).LE.0.0) RETURN
      DO 311 I=1,NHAT
      ALGEN(I)=ALGEN(I) + (LOSSI(1)/HORRAN)
  311 CONTINUE
      RETURN
      END
C
      REAL FUNCTION INTEG1(ALPHA,BETA,ZHAT,A)
C
      F(X) = (SIN(X)/COS(X)**2) + ALOG((1.0+SIN(X))/COS(X))
C
C        EVALUATE THE INTEGRAL RESULTING FROM SOLUTION OF EQUATION
C        RELATING LOSS(THETA) AND ATTENUATION ALPHA(Z)
C
      F1 = ALPHA**2
      F2 = BETA**2
      F3 = ZHAT**2
      F4 = SQRT(F3-F1)/A
      F5 = SQRT(F3-F2)/A
      XLO = ATAN(F5)
      XHI = ATAN(F4)
      INTEG1 = A*A*(F(XHI)-F(XLO))
      RETURN
      END
C
      REAL FUNCTION INTEG2(ALPHA,BETA,ZHAT,A)
```

```
C
C         EVALUATE THE INTEGRAL RESULTING FROM SOLUTION OF EQUATION
C         RELATING LOSS(THETA) AND ATTENUATION ALPHA(Z)
C
      F1 = ZHAT**2
      F2 = BETA**2
      F3 = ALPHA**2
      F4 = A*A
      F5 = F1 + F4
      F6 = SQRT(F5)
      DEN1 = AMAX1(1.0E-10,SQRT(F1-F2))
      DEN2 = SQRT(F1-F3)
      CALL ELI2(RESU,BETA/DEN1,A/F6,1.0,F4/F5)
      CALL ELI2(RESL,ALPHA/DEN2,A/F6,1.0,F4/F5)
      INTEG2 = 2.0 * F6 * (RESU - RESL)
      RETURN
      END
C
      REAL FUNCTION LOFZ(THETI,LOSSI,NI,ZHAT,HORRAN)
      REAL THETI(NI) , LOSSI(NI)
C
C         EVALUATE THE TABULATED LOSS FUNCTION L(THETA) AT AN INCIDENT
C         ANGLE THETA CORRESPONDING TO THE TURNING DEPTH ZHAT
C         COMPUTE THETA(Z) BY USING FUNCTIONAL FORM OF PROFILE AND SNELLS
C         LAW.
C
C         WE GET A RELATION FOR THE INCIDENT ANGLE THETA(PARABOLIC)
C         BY COMBINING THE FUNCTIONAL FORM OF THE PROFILE AND THE SNELLS
C         LAW, BOTH IN THE PARABOLIC APPROXIMATION. HOWEVER, THE LOSS
C         FUNCTION L(THETA) IS GIVEN AS A FUNCTION OF THETA(ELLIPTIC) ...
C         THE PHYSICAL ANGLE. CONVERT VIA THE FOLLOWING
C                   SIN(THETA-ELLIPTIC) = TAN(THETA-PARABOLIC)
C
       THETZ = (180.0/3.1415926) * ARSIN(3.1415926*ZHAT/HORRAN)
C         USE TABLE LOOKUP AND LINEAR INTERPOLATION IN THE GIVEN
C         LOSS FUNCTION. INDEPENDANT VARIABLE THETA IS INCIDENT ANGLE
      DO 10 I = 2 , NI
      IF ( THETZ .GT. THETI(I) ) GO TO 10
      LOFZ = LOSSI(I-1) + (LOSSI(I)-LOSSI(I-1)) * (THETZ -THETI(I-1))
     A  / (THETI(I)-THETI(I-1))
      RETURN
   10 CONTINUE
C         ERROR EXIT ... ANGLE IS NOT IN TABULATED ANGLE RANGE
      WRITE(6,11) THETZ , THETI(1) , THETI(NI)
   11 FORMAT(41H0LOSS FUNCTION ANGLE NOT IN TABLE RANGE     ,3F10.3)
      STOP 007
      END
C
```

```
C*********************************************************************
C
C         PURPOSE ... COMPUTE THE GENERALIZED ELLIPTIC INTEGRAL OF THE
C                     SECOND KIND
C
C         USAGE ...   CALL ELI2(R,X,CK,A,B)
C
C         DESCRIPTION OF PARAMETERS
C              R      RESULT
C              X      UPPER INTEGRATION BOUND
C              CK     COMPLEMENTARY MODULUS (1.0-K*K)
C              A,B    CONSTANTS IN THE INTEGRAND
C         SPECIAL CASE ...
C                     E(ATAN(X),K) OBTAINED WITH A = 1.0,B = CK*CK
C
C         SUBROUTINES REQUIRED ... NONE
C
C         METHOD      SERIES SUMMATION WITH LANDEN TRANSFORMATION
C
C         SEE IBM LIBRARY SSP
C
C*********************************************************************
C         TEST ARGUMENT
      SUBROUTINE ELI2(R,X,CK,A,B)
      IF(X) 2,1,2
    1 R = 0.0
      RETURN
C         TEST MODULUS
    2 C = 0.0
      D = 0.5
      IF(CK) 7,3,7
    3 R = SQRT(1.0 + X*X)
      R=(A-B)*ABS(X)/R+B*ALOG(ABS(X)+R)
C         TEST SIGN OF ARGUMENT
    4 R = R+C*(A-B)
      IF(X) 5,6,6
    5 R = - R
    6 RETURN
C         INITIALIZATION
    7 AN =  (B+A)*0.5
      AA = A
      R = B
      ANG = ABS(1.0/X)
      PIM = 0.0
      ISI = 0
      ARI = 1.0
      GEO = ABS(CK)
C         LANDEN TRANSFORMATION
```

```
    8 R = AA * GEC + R
      SGEO = ARI * GEO
      AA = AN
      AARI = ARI
C         ARITHMETIC MEAN
      ARI = GEO + ARI
C         SUM OF SINE VALUES
      AN = (R/ARI+AA)*0.5
      AANG = ABS(ANG)
      ANG = -SGEC / ANG + ANG
      PIMA = PIM
      IF(ANG) 10,9,11
    9 ANG = -1.0E-8*AANG
   10 PIM = PIM + 3.14159265
      ISI = ISI + 1
   11 AANG = ARI*ARI + ANG*ANG
      P = D / SQRT(AANG)
      IF(ISI-4) 13,12,12
   12 ISI = ISI - 4
   13 IF(ISI-2) 15,14,14
   14 P = - P
   15 C = C + P
      D = D * (AARI - GEO) * 0.5 / ARI
      IF(ABS(AARI-GEO)-1.0E-6*AARI) 17,17,16
   16 SGEO = SQRT(SGEO)
C         GEOMETRIC MEAN
      GEO = SGEO + SGEO
      PIM = PIM + PIMA
      ISI = ISI + ISI
      GC TO 8
C         ACCURACY WAS SUFFICIENT
   17 R = (ATAN(ARI/ANG)+PIM)*AN/ARI
      C = C + D*ANG/AANG
      GC TO 4
      END
      SUBROUTINE RDPROF(NPROF)
      CCMMON/EARTH/ISPH
      CCMMON /UNITS/ LC,LP,IT
      DIMENSION Z(50),C(50),Z1(50),C1(50)
C
C*********
C
C         THIS ROUTINE READS PROFILES FROM CARDS AND FAKES A CFIELD TAPE
C
C*********
      IF(ISPH.LE. 0) WRITE(LP,12)
   12 FORMAT(40HOSPHERICAL EARTH CORRECTION APPLIED      ,/)
C
```

```
      REWIND 2
      CVP= 1.0E10
C
      DC 100 I = 1 , NPROF
      READ(LC,1) RANGE,NPTS
      RANGE1=RANGE
C         CONVERT INPUT RANGE (IN NMILES) TO FEET
      RANGE = RANGE * 6076.1
      IF(I.NE.1) WRITE(2) RANGE
    1 FORMAT(F10.2,I5)
      READ(LC,2)  (Z(L),C(L),L=1,NPTS)
    2 FORMAT(8F10.2)
      IF(C(1).GT.3000.0) GO TO 6
C         CONVERT INPUT TO ENGLISH UNITS
      DO 3 J=1,NPTS
      Z1(J)=Z(J)
      Z(J)=Z(J)/.3048
      C1(J)=C(J)
      C(J)=C(J)/.3048
    3 CCNTINUE
      WRITE(LP,4) RANGE1
    4 FCRMAT(//,22HOPROFILE AT RANGE =          ,F10.2,3H NM     ,/,
     X 10H DEPTH(FT),5X,10HC(Z) (FT/S),5X,10H   DEPTH(M),5X,10H C(Z) (M/S))
      WRITE(LP,5) (Z(J),C(J),Z1(J),C1(J),J=1,NPTS)
      GC TO 90
    5 FORMAT(F10.C,F15.2,F15.0,F15.2)
    6 CCNTINUE
      WRITE(LP,41) RANGE1
   41 FCRMAT(//,23HOPROFILE AT RANGE            ,F10.2,3H NM,/,
     X 10H DEPTH(FT),5X,10HC(Z) (FT/S)       )
      WRITE(LP,7) (Z(J),C(J),J=1,NPTS)
    7 FCRMAT(F10.0,F15.2)
   90 CCNTINUE
      IF(ISPH.GT.0) GO TO 10
      DO 11 J = 1 , NPTS
      ZTEMP = Z(J)
      Z(J)=Z(J)*(1.0+Z(J)/4.1807E7)
      C(J)=C(J)*2.09035E7/(2.09037E7-ZTEMP)
   11 CONTINUE
   10 CONTINUE
C
C         FIND MIN
C
      IF(I.GT.1) GO TO 9
      DC 8 J = 1 , NPTS
      CVP = AMIN1(CVP,C(J))
    8 CCNTINUE
      WRITE(2) CVP
```

```
      WRITE(2) CVP
    9 CONTINUE
C
C        DUMP TO TAPE
C
      WRITE(2) NPTS,(Z(L),C(L),L=1,NPTS)
  100 CONTINUE
      RANGE=1.0E10
      WRITE(2) RANGE
      REWIND 2
C
      RETURN
C
      END
C
      SUBROUTINE GETBOT(IFLAT,DMAX)
C
C
C        READ AND PRINT BATHYMETRY. CONVERT RANGE,DEPTH TO FEET
C
C
      COMMON /UNITS/ LC,LP,IT
      COMMON /BATHY/ RE,KB,NB,BR(101),BZ(101)
      DATA FNM,FT,RAD/6076.1,0.3048,0.174532925199943E-01/
C
      NB=IABS(IFLAT)
C
      IF(NB.LE.100) GO TO 10
C
      WRITE(LP,900)
900   FORMAT(41H0***BOTTOM DATA EXCEED AVAILABLE STORAGE.)
      STOP
C
10    IF (NB.EQ.0) GO TO 30
C
      READ(LC,910) (BR(I),BZ(I),I=1,NB)
910   FORMAT(8F10.2)
C
      WRITE(LP,920)
920   FORMAT(1H0,7X,10HBATHYMETRY/
     1        24H POINT    RANGE      DEPTH)
C
      DO 20 I=1,NB
      WRITE(LP,930) I,BR(I),BZ(I)
930   FORMAT(2X,I3,F8.1,3X,F8.1)
      BR(I)=FNM*BR(I)
      IF (IFLAT.LT.0) BZ(I)=BZ(I)/FT
20    DMAX=AMAX1(BZ(I),DMAX)
```

```
C
      EZ(NB+1)=EZ(NB)
C
   30 CCNTINUE
      ER(NB+1)=1.CE16
      RETURN
      END
```

APPENDIX B: WT SOURCE CODE

```
C      THE PURPOSE OF THIS PROGRAM IS TO TRANSFORM THE ACOUSTIC
C      SSFFT PARABOLIC EQUATION MODEL  "PRESSURE DATA" INTO THE
C      WAVENUMBER DOMAIN FOR FURTHER ANALYSIS BY THE WAVENUMBER
C      TECHNIQUE AS DESCRIBED BY RICHARD LAUER OF NORDA
C
C
       INTEGER M
       INTEGER TITLE(16)
       REAL*4 Q(2050),P(2050),QMOD(2050)
       DIMENSION PREPHA(2050),IWK(13200),WK(13200)
       DIMENSION PR(2050),PI(2050),RANGE(2050)
       DIMENSION D(21),BUFO(21)
       COMPLEX PREPHA
       PIE=3.141592654
       ATC=0.0015
       NTOT = 2050
C
C      LT IS THE MISC. DATA INPUT FILE FROM THE PEMODEL
C
       LT = 3
C
C      LT1 IS THE RANGE AND PRESSURE DATA INPUT FILE FROM THE PEMODEL
C
       LT1 = 4
C
C      LO IS THE DATA OUTPUT FILE OF WAVENUMBER DATA
C
       LO = 7
C
C      LD ID THE DIAGNOSTIC OUTPUT FILE
C
       LD = 8
       REWIND LT
       REWIND LT1
C
C      READ INPUT
C
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
       READ(LT,800) (TITLE(I),I=1,16)
   800 FORMAT(16A4)
C
C      FREQ    INPUT FREQUENCY
C      ZS      SOURCE DEPTH
C      NPT     NUMBER OF POINTS IN THE "PRESSURE" ARRAY
C      ND      NUMBER OF RECEIVER DEPTHS
C      CLMIN   (NOT USED) SEE PE PROGRAM
C      DCL     (NOT USED) SEE PE PROGRAM
C      FACT    (NOT USED) SEE PE PROGRAM
```

```
C       IPLOT    (NOT USED) SEE PE PROGRAM
C
        READ(LT,810) FREQ,ZS,NPT,ND,CLMIN,DCL,FACT,IPLOT
  810   FORMAT(2(2X,F10.2),2(2X,I4),3(2X,F7.2),2X,I3)
        WRITE(LD,501)
        WRITE(LD,*)FREQ,ZS,NPT,ND
C
C       READ-IN THE RECEIVER DEPTHS
C
        READ(LT,820) (D(I),I=1,ND)
        WRITE(LD,*)D(1)
  820   FORMAT(5E15.7)
C
C       DDDD     GRID DEPTH DIRECTLY BELOW THE SOURCE DEPTH
C       FK       REFERENCE WAVENUMBER (AS DETERMINED FROM MINIMUM
C                SOUND SPEED IN SOURCE PROFILE
C       VELC     REFERENCE SOUND SPEED
C       ATTEN    VOLUME ATTENUATION IN THE WATER MASS
C       DMAX     MAXIMUM WATER DEPTH
C
        READ(LT,507) DDDD,FK,VELC,ATTEN,DMAX
  507   FORMAT(5(1X,E15.7))
C
C       BEAM     HALF BEAM WIDTH OF THE SOURCE SIGNAL
C       DRHOLD   (NOT USED) SEE PE PROGRAM
C       IFLAGU   (NOT USED) SEE PE PROGRAM
C
        READ(LT,508)BEAM,DRHOLD,IFLAGU
  508   FORMAT(2(1X,E15.7),1X,I1)
        ATC = ATTEN
C
C       ZERO OUT RANGE ARRAY AND LIMIT INPUT TO PRECLUDE PROGRAM
C       CRASH
C
         DO 825 I=1,NTOT
  825   RANGE(I) = 0.0
        IF(NPT .GT. 2050) NPT = 2050
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
  501   FORMAT(' ','PASSED POINT 1')
        WRITE(LD,*)DDDD,FK,VELC,ATTEN,DMAX
        WRITE(LD,509) DDDD
  509   FORMAT(' ','FIELD DEPTH IS',F10.3)
C
C       DETERMINE TRANSFORM SIZE
C
  525   DO 111 I=1,12
        NN=2**I
        IF(NN.GT.NPT) GO TO 113
```

```
  111 CONTINUE
  113 CONTINUE
C
C     READ-IN RANGE AND COMPLEX "PRESSURE" DATA
C
      DO 860 IWT2=1,NPT
      READ (LT1,850) RANGE(IWT2),PR(IWT2),PI(IWT2)
  850 FORMAT(3(2X,E15.7))
  860 CONTINUE
      WRITE(LD,901)
  901 FORMAT('0','PASSED POINT 2')
C
C     *********************************
C
C
C     LET RANGE INCREMENT EQUAL THE FIRST RANGE STEP
C
      RINC=RANGE(2) - RANGE(1)
C
C     CONVERT NAUTICAL MILES TO FEET
C
      DELR=RINC*6076.1
C
C     ZERO OUT FUTURE ARRAYS
C     QMOD WILL BE BETA
C
      DO 60  I=1,NTOT
      PREPHA(I)=CMPLX(0.0,0.)
      P(I)=0.0
      Q(I)=0.0
      QMOD(I)=0.0
   60 CONTINUE
      WRITE(LD,903)
  903 FORMAT('0','PASSED POINT 3')
      DO 75 I=1,NPT
C
C     INSERT THE HANKEL APPROXIMATION TO OBTAIN TRUE PRESSURE FROM
C     INPUT "PRESSURE" AND BACK OUT ATTENUATION
C
      P(I)=(PR(I)*COS(FK*RANGE(I))-PI(I)*SIN(FK*RANGE(I)))
     1*EXP(ATC*RANGE(I))
      Q(I)=(PR(I)*SIN(FK*RANGE(I))+PI(I)*COS(FK*RANGE(I)))
     1*EXP(ATC*RANGE(I))
   75 CONTINUE
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
      WRITE(LD,904)
  904 FORMAT('0','PASSED POINT 4')
C
C     LOAD REAL AND IMAGINARY PARTS INTO COMPLEX ARRAY
```

```
C
      DO 100 I=1,NN
      PREPHA(I)=CMPLX(P(I),Q(I))
  100 CONTINUE
      WRITE(LD,905)
  905 FORMAT('0','PASSED POINT 5')
C
C     FAST FOURIER TRANSFORM PRESSURE DATA
C
      CALL FFTCC(PREPHA,NN,IWK,WK)
      WRITE(LD,906)
  906 FORMAT('0','PASSED POINT 6')
C
C     XI      IS INITIAL HORIZONTAL WAVENUMBER
C     DZ      IS WAVENUMBER INCREMENT
C
C     MULTIPLICATION BY 1000 FOR COMPUTER USAGE ONLY
C
      XI=(2.0*PIE*FREQ/VELC-2.0*PIE/DELR)*1000.0
      DZ=(2.0*PIE/(NN*DELR))*1000.0
      WRITE(LD,957)
  957 FORMAT('0','PASSED POINT 6A')
C
C     Q WILL BECOME THE SPECTRUM HORIZONTAL WAVENUMBER (HORIZ AXIS)
C     P AT END OF LOOP WILL BE SPECTRAL INTENSITY (VERT AXIS)
C
      DO 125 I=1,NN
      RN=FLOAT(I)
      A=XI+RN*DZ
      P(I)=AIMAG(PREPHA(I))
      Q(I)=REAL(PREPHA(I))
      P(I)=(P(I)**(2)+Q(I)**(2))
C
C     1000 FOR COMPUTER USAGE ONLY, IS REMOVED
C
      Q(I)=A/1000.
  125 CONTINUE
      WRITE(LD,907)
  907 FORMAT('0','PASSED POINT 7')
      PMAX=P(1)
C
C     DETERMINE MAXIMUM P FOR NORMALIZATION IN VERT AXIS
C
      DO 127 I=1,NN
      IF(P(I).GT.PMAX)  PMAX=P(I)
  127 CONTINUE
C
C     NORMALIZE VERT AXIS
```

```
C
      DO 101 I=1,NN
      P(I)=P(I)/PMAX
  101 CONTINUE
      WRITE(LD,908)
  908 FORMAT('0','PASSED POINT 8')
C@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
C
C     QMAX    IS THE REFERENCE WAVENUMBER
C     SCALE HORIZONTAL AXIS IN BETA (QMOD)
C
      QMAX=FK
      DO 128 I=1,NN
      QMOD(I)=SQRT(ABS((QMAX**2)-(Q(I)**2)))
  128 CONTINUE
      WRITE(LD,909)
  909 FORMAT('0','PASSED POINT 9')
      DO 223 I = 1,NN
C
C     OUTPUT TO PLOTTING PROGRAM (PEFFTPT)
C
  223 WRITE(LO,224)  Q(I),QMOD(I),P(I)
  224 FORMAT(3(2X,E15.7))
      STOP
      END
```

APPENDIX C: WT GRAPHICS SOURCE CODE

```
C######################################################################
C#################### START OF EXECUTABLE CODE #########################
      INTEGER TITLE(15)
      DIMENSION D(21),BUFO(21)
      DIMENSION C(2050),QMOD(2050),P(2050)
      DIMENSION RANGE(2050)
      DATA  IY/'Y'/,NO/'N'/
C
C     LT IS THE MISC. DATA INPUT FILE FROM THE PEMODEL
C
      LT = 3
C
C     LT1 IS THE RANGE DATA INPUT FILE FROM THE PEMODEL
C
      LT1 = 4
C
C     LO IS THE DATA INPUT FILE OF WAVENUMBER DATA FROM THE FFT
C
      LO = 7
C
C     LD ID THE DIAGNOSTIC OUTPUT FILE
C
      LD = 8
   10 CONTINUE
      REWIND LT
      REWIND LT1
      REWIND LO
C
C     READ INPUT
C
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
      READ(LT,100) (TITLE(I),I=1,15)
  100 FORMAT(15A4)
      READ(LT,110) FREQ,ZS,NPT,ND,CLMIN,DCL,FACT,IPLOT
  110 FORMAT(2(2X,F10.2),2(2X,I4),3(2X,F7.2),2X,I3)
      WRITE(LD,150)
      READ(LT,120) (D(I),I=1,ND)
  120 FORMAT(5E15.7)
      IF(NPT .GT. 2050) NPT = 2050
      DO 140 I=1,NPT
      READ(LT1,120) RANGE(I)
  130 FORMAT(2X,E15.7)
  140 CONTINUE
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
  150 FORMAT(' ','PASSED POINT 1')
      READ(LT,160) DDDD,FK,VELC,ATTEN,DMAX
  160 FORMAT(5(1X,E15.7))
      WRITE(LD,170) DDDD
```

```
  170 FORMAT(' ','FIELD DEPTH IS',F10.3)
      DO 180 I=1,12
      NN=2**I
      IF(NN.GT.NPT) GO TO 190
  180 CONTINUE
  190 CONTINUE
C ****CHANGE TRANSFORM SIZE 8=256 9=512 10=1024 11=2048 12=4096
CCCCC N = I
      N = I - 1
C ********************
      WRITE(LD,215) NN,N
  215 FORMAT(' ','NN IS ',I4,5X,'N IS ',I4)
      NN=2**N
      DO 220 I = 1,NN
  220 READ (LO,230) Q(I),QMOD(I),P(I)
  230 FORMAT(3(2X,E15.7))
      WRITE(LD,235)
  235 FORMAT(' ','INPUT OF DATA HAS BEEN COMPLETED')
      CALL COMPRS
C *****
      PMIN = P(1)
      PMAX = P(1)
      QMMIN = QMOD(1)
      QMMAX = QMOD(1)
      QMIN = Q(1)
      QMAX = Q(1)
      DO 240 I = 2,NN
      IF(Q(I) .GT. QMAX) QMAX = Q(I)
      IF(Q(I) .LT. QMIN) QMIN = Q(I)
      IF(P(I) .GT. PMAX) PMAX = P(I)
      IF(P(I) .LT. PMIN) PMIN = P(I)
      IF(QMOD(I) .GT. QMMAX) QMMAX = QMOD(I)
  240 IF(QMOD(I) .LT. QMMIN) QMMIN = QMOD(I)
C
C     CLEAR SCREEN AT THE TERMINAL
C
      CALL FRTCMS('CLRSCRN ')
C
C
C     *** GET INCREMENT OF X-AXIS
          WRITE(6,900) QMMIN,QMMAX
 900  FORMAT(/,' THE CURRENT MINIMUM X-AXIS (BETA) VALUE IS '
     1,E9.2,/,
     *' AND THE MAXIMUM X-AXIS VALUE IS ',E9.2,/,' ENTER NEW',
     *' MINIMUM AND MAXIMUM VALUES OR A ZERO FOR EACH IF YOU',/,
     *' WOULD LIKE TO ACCEPT THE CURRENT VALUES',/,
     * ' ENTER DESIRED X-AXIS INCREMENT:',
     */,' (OR 0 FOR SELF SCALING)')
```

```
          READ(5,*) XX1,XX3,X2
          IF(XX1 .NE. 0.0) QMMIN = XX1
          IF(XX3 .NE. 0.0) QMMAX = XX3
C
C     CLEAR SCREEN AT THE TERMINAL
C
      CALL FRTCMS('CLRSCRN ')
          WRITE(6,901) QMIN,QMAX
 901  FORMAT(/,' THE CURRENT MINIMUM X-AXIS (WAVENUMBER)',
     1' VALUE IS ',E9.2,/,
     *' AND THE MAXIMUM X-AXIS VALUE IS ',E9.2,/,' ENTER NEW',
     *' MINIMUM AND MAXIMUM VALUES OR A ZERO FOR EACH IF YOU',/,
     *' WOULD LIKE TO ACCEPT THE CURRENT VALUES',/,
     * ' ENTER DESIRED X-AXIS INCREMENT:',
     */,' (OR 0 FOR SELF SCALING)')
          READ(5,*) XX1,XX3,XX2
          IF(XX1 .NE. 0.0) QMIN = XX1
          IF(XX3 .NE. 0.0) QMAX = XX3
C
C     CLEAR SCREEN AT THE TERMINAL
C
      CALL FRTCMS('CLRSCRN ')
C     *** GET INCREMENT OF Y-AXIS
          WRITE(6,903)PMIN,PMAX
 903  FORMAT(/,' THE CURRENT MINIMUM Y-AXIS (MAGNITUDE)',
     1' VALUE IS ',E9.2,/,
     *' AND THE MAXIMUM Y-AXIS VALUE IS ',E9.2,/,' ENTER NEW',
     *' MINIMUM AND MAXIMUM VALUES OR A ZERO FOR EACH IF YOU',/,
     *' WOULD LIKE TO ACCEPT THE CURRENT VALUES',/,
     * ' ENTER DESIRED Y-AXIS INCREMENT:',
     */,' (OR 0 FOR SELF SCALING)')
          READ(5,*) Y11,Y33,Y2
          IF(Y11 .NE. 0.0) PMIN = Y11
          IF(Y33 .NE. 0.0) PMAX = Y33
      CALL PAGE(11.,8.5)
      CALL INTAXS
      CALL SETDEV(1,0)
      CALL GRACE(0.0)
      CALL NOBRDR
      CALL AREA2D(9.0,6.0)
      CALL XNAME('SCALED WAVENUMBER BETA (1/FT)',29)
      CALL YNAME('NORMALIZED SPECTRAL INTENSITY ',30)
      CALL YTICKS(5)
      CALL XTICKS(5)
      CALL HEADIN(TITLE,60,1.5,1)
      IF((X2 .NE. 0.0).AND.(Y2 .NE. 0.0))
     *CALL GRAF(QMMAX,XX2,QMMIN,PMIN,Y2,PMAX)
      IF((X2 .EQ. 0.0).AND.(Y2 .EQ. 0.0))
```

```
     *CALL GRAF(QMMAX,'SCALE',QMMIN,PMIN,'SCALE',PMAX)
      IF((X2 .EQ. 0.0).AND.(Y2 .NE. 0.0))
     *CALL GRAF(QMMAX,'SCALE',QMMIN,PMIN,Y2,PMAX)
      IF((X2 .NE. 0.0).AND.(Y2 .EQ. 0.0))
     *CALL GRAF(QMMAX,XX2,QMMIN,PMIN,'SCALE',PMAX)
      CALL BLSYM
      CALL SPLINE
      CALL CURVE(QMOD,P,NN,0)
      CALL MESSAG('SOURCE DEPTH $',100,-.35,6.7)
      CALL REALNC(ZS,2,'ABUT','ABUT')
      CALL MESSAG(' FT., RECEIVER DEPTH $',100,'ABUT','ABUT')
      CALL REALNC(D(1),2,'ABUT','ABUT')
      CALL MESSAG(' FT., FREQUENCY $',100,'ABUT','ABUT')
      CALL REALNC(FREQ,2,'ABUT','ABUT')
      CALL MESSAG(' HZ$',100,'ABUT','ABUT')
      CALL MESSAG('WATER DEPTH $',100,-.35,6.5)
      CALL REALNC(DMAX,2,'ABUT','ABUT')
      CALL MESSAG(' FT., FIELD DEPTH $',100,'ABUT','ABUT')
      CALL REALNC(DDDD,2,'ABUT','ABUT')
      CALL MESSAG(' FT., ATTENUATION COEF $',100,'ABUT','ABUT')
      CALL REALNC(ATTEN,-3,'ABUT','ABUT')
      CALL MESSAG('AVERAGE WAVE NUMBER $',100,-.35,6.3)
      CALL REALNC(FK,-3,'ABUT','ABUT')
      CALL MESSAG(', REFERENCE SOUND SPEED $',100,'ABUT','ABUT')
      CALL REALNC(VELC,2,'ABUT','ABUT')
      CALL MESSAG(' FT./SEC$',100,'ABUT','ABUT')
      CALL DCT
      CALL GRID(1,1)
      CALL ENDPL(0)
C *****
C ********************************
C ***********************************
C *****
      CALL PAGE(11.,8.5)
      CALL INTAXS
      CALL SETDEV(1,0)
      CALL GRACE(0.0)
      CALL NOBRDR
      CALL AREA2D(9.0,6.0)
      CALL XNAME('HORIZONTAL WAVENUMBER$',100)
      CALL YNAME('NORMALIZED SPECTRAL INTENSITY ',30)
      CALL YTICKS(5)
      CALL XTICKS(5)
      CALL RESET('DOT')
      CALL HEADIN(TITLE,60,1.5,1)
      IF((XX2 .NE. 0.0).AND.(Y2 .NE. 0.0))
     *CALL GRAF(QMAX,XX2,QMIN,PMIN,Y2,PMAX)
      IF((XX2 .EQ. 0.0).AND.(Y2 .EQ. 0.0))
```

```
     *CALL GRAF(QMAX,'SCALE',QMIN,PMIN,'SCALE',PMAX)
      IF((XX2 .EQ. 0.0) .AND. (Y2 .NE. 0.0))
     *CALL GRAF(QMAX,'SCALE',QMIN,PMIN,Y2,PMAX)
      IF((XX2 .NE. 0.0) .AND. (Y2 .EQ. 0.0))
     *CALL GRAF(QMAX,XX2,QMIN,PMIN,'SCALE',PMAX)
      CALL BLSYM
      CALL SPLINE
      CALL CURVE(Q,P,NN,0)
      CALL MESSAG('SOURCE DEPTH $',100,-.35,6.7)
      CALL REALNC(ZS,2,'ABUT','ABUT')
      CALL MESSAG(' FT., RECEIVER DEPTH $',100,'ABUT','ABUT')
      CALL REALNC(D(1),2,'ABUT','ABUT')
      CALL MESSAG(' FT., FREQUENCY $',100,'ABUT','AEUT')
      CALL REALNC(FREQ,2,'ABUT','ABUT')
      CALL MESSAG(' HZ$',100,'ABUT','ABUT')
      CALL MESSAG('WATER DEPTH $',100,-.35,6.5)
      CALL REALNC(DMAX,2,'ABUT','ABUT')
      CALL MESSAG(' FT., FIELD DEPTH $',100,'ABUT','ABUT')
      CALL REALNC(DDDD,2,'ABUT','ABUT')
      CALL MESSAG(' FT., ATTENUATION COEF $',100,'ABUT','ABUT')
      CALL REALNC(ATTEN,-3,'ABUT','ABUT')
      CALL MESSAG('AVERAGE WAVE NUMBER $',100,-.35,6.3)
      CALL REALNC(FK,-3,'ABUT','ABUT')
      CALL MESSAG(', REFERENCE SOUND SPEED $',100,'ABUT','ABUT')
      CALL REALNC(VELC,3,'ABUT','ABUT')
      CALL MESSAG(' FT./SEC$',100,'ABUT','ABUT')
      CALL DOT
      CALL GRID(1,1)
      CALL ENDPL(0)
C
C     CLEAR SCREEN AT THE TERMINAL
C
      CALL FRTCMS('CLRSCRN ')
C     *** WOULD YCU LIKE TO MAKE ANOTHER SET OF PLOTS?
 1800     WRITE(6,1900)
 1900     FORMAT(/,' WOULD YOU LIKE TO MAKE ANOTHER SET OF PLOTS',/,
     *    ' ANSWER (Y/N)')
          READ(5,905) IANS
  905 FCRMAT(1A1)
      IF((IANS .NE. IY) .AND. (IANS .NE. NO)) GO TO 1800
          IF ( IANS.EQ.IY ) GO TO 10
      CALL DCNEPI
      STOP
      END
```

APPENDIX D: IFD SOURCE CODE (AT NPS)

```
C     THIS SUBROUTINE IS USED TO OUTPUT DATA IN A FORMAT WHICH IS
C     REDUCED IN VOLUME AND COMPATIBLE FOR THE WAVENUMBER TECHNIQUE
C     (WT)
C
C
C
C     ******************************************************************
C     ******************************************************************
C     *  IMPLICIT FINITE-DIFFERENCE PROGRAM FOR                        *
C     *  SOLVING THE PARABOLIC EQUATION                                *
C     ******************************************************************
C     ******************************************************************
C     *  LT LARRY JAEGER                                               *
C     *  U.S. NAVAL POSTGRADUATE SCHOOL                                *
C     *  MONTEREY, CA  93943                                           *
C     ******************************************************************
C     ******************************************************************
C     ******************************************************************
C     *** ALPHABETICAL LIST OF PROGRAM VARIABLES FOLLOWS:              *
C     ******************************************************************
C     *** A      - ARRAY - COEFFICIENT A IN PARABOLIC EQUATION
C                  (IN WATER)
C     *** A2     - COEFFICIENT A IN PARABOLIC EQUATION (IN SEDIMENT)
C     *** ALPHA  - VOLUME ATTENUATION - DB/METER
C     *** ATT    - ARRAY - ATTENUATION COEFFICIENT FOR ARTIFICIAL
C                  ATTENUATION LAYER
C     *** BEDA1  - BETA 1 AS DEFINED IN LEE AND MCDANIEL (1983)
C     *** BEDA2  - BETA 2 AS DEFINED IN LEE AND MCDANIEL (1983)
C     *** BETA1  - ATTENUATION IN WATER - DB/WAVELENGTH
C     *** BETA2  - ATTENUATION IN SEDIMENT - DB/WAVELENGTH
C     *** BR     - ARRAY - RANGE FOR BOTTOM PROFILE - METERS
C     *** BZ     - ARRAY - DEPTH FOR BOTTOM PROFILE - METERS
C     *** C      - ARRAY - CONTAINS TRIDIAGONAL MATRIX SYSTEM THAT NEEDS
C                  TO BE SOLVED (SEE SUBROUTINE TRIDG)
C     *** C0     - REFERENCE SOUND SPEED - METERS/SEC
C     *** C2     - SOUND SPEED IN SEDIMENT
C     *** COSE   - COS ( THETA )
C     *** CR     - ARRAY - STORAGE SPACE USED IN SUBROUTINES TRIDG AND
C                  TRIDL
C     *** CSVP   - ARRAY - SOUND SPEED IN SOUND SPEED PROFILE -
C                  METERS/SEC
C     *** CTWO   - ARRAY - STORAGE SPACE USED IN SUBROUTINES TRIDG AND
C                  TRIDL
C     *** CWATER- ARRAY - SOUND SPEED AT GRID POINTS IN WATER COLUMN
C     *** DELTA  - DELTA AS DEFINED IN LEE AND MCDANIEL (1983)
C     *** DELIN  - 1 / DELTA
C     *** DR     - RANGE STEP - METERS
C     *** DRIVL  - RANGE STEP ALONG LEVEL INTERFACE - METERS
```

```
C     *** DRMAX - MAXIMUM ALLOWABLE RANGE STEP - METERS
C     *** DZ    - DEPTH INCREMENT OF SOLUTION - METERS
C     *** EYE   - COMPLEX "I"
C     *** FRQ   - FREQUENCY - HZ
C     *** GAMMA1- GAMMA 1 AS DEFINED IN LEE AND MCDANIEL (1983)
C     *** GAMMA2- GAMMA 2 AS DEFINED IN LEE AND MCDANIEL (1983)
C     *** IBOT1 - POINTER THAT POINTS TO BOTTOM PROFILE POINT AT START
C                 OF BOTTOM SEGMENT
C     *** IBOT2 - POINTER THAT POINTS TO BOTTOM PROFILE POINT AT END
C                 OF BOTTOM SEGMENT
C     *** ID    - ARRAY - RUN IDENTIFICATION
C     *** IFACE - POINTER THAT POINTS TO INTERFACE AT RANGE RA1
C     *** IFACE2- POINTER THAT POINTS TO INTERFACE AT RANGE RA2
C     *** IFACEM- IFACE - 1
C     *** IFACEP- IFACE + 1
C     *** IPZ   - EVERY IPZ'TH VALUE IN DEPTH IS PRINTED
C     *** ISLOPE- SLOPE FLAG:
C                 ISLOPE = 1  -  BOTTOM SLOPES DOWN
C                 ISLOPE = 2  -  BOTTOM LEVEL
C                 ISLOPE = 3  -  BOTTOM SLOPES UP
C                 ISLOPE = 4  -  BOTTOM SLOPES DOWN, BOTTOM MODIFIED
C                 ISLOPE = 5  -  BOTTOM SLOPES UP, BOTTOM MODIFIED
C     *** ITEMP - TEMPORARY VARIABLE
C     *** IWZ   - GRID POINT CORRESPONDING TO RECEIVER DEPTH
C     *** N     - NUMBER OF EQUI-SPACED GRID POINTS IN U
C               - INCLUDES BOTTOM POINT - DOES NOT INCLUDE SURFACE POINT
C     *** NA    - NUMBER OF POINTS IN ARTIFICIAL ATTENUATION LAYER
C     *** NBOT  - NUMBER OF POINTS IN BOTTOM PROFILE (BR AND BZ)
C     *** NIU   - UNIT NUMBER FOR INPUT DATA
C     *** NM1   - N - 1
C     *** NOU   - UNIT NUMBER FOR OUTPUT PLOTTER FILE
C     *** NPOUT - UNIT NUMBER FOR OUTPUT PRINTER FILE
C     *** NSTEP - NUMBER OF RANGE STEPS ALONG A BOTTOM SEGMENT
C     *** NSTEP1- NUMBER OF RANGE STEPS CORRESPONDING TO ONE VERTICAL
C                 GRID STEP FOR MODIFIED BOTTOM
C     *** NSVP  - NUMBER OF POINTS IN CSVP AND ZSVP
C     *** NWMAX - NUMBER OF GRID POINTS IN WATER AT MAX DEPTH
C     *** NXLFS - NUMBER OF NEXT LEVEL SECTION FOLLOWING A SLOPING
C                 SECTION FOR A MODIFIED BOTTOM
C     *** PDR   - RANGE INCREMENT AT WHICH SOLUTION IS PASSED TO
C                 OUTPUT PRINTER FILE - METERS
C     *** PDZ   - DEPTH INCREMENT AT WHICH SOLUTION IS PRINTED - METERS
C     *** PI    - THE VALUE OF PI
C     *** PL    - PROPAGATION LOSS - DB
C     *** PRTIN - INITIAL RANGE FOR PLOT AND PRINT FILE - METERS
C     *** PRTOT - FINAL RANGE FOR PLOT AND PRINT FILE - METERS
C     *** R1    - RANGE AT WHICH BOTTOM DEPTH IS AVAILABLE - METERS
C     *** R2    - NEXT RANGE AT WHICH BOTTOM DEPTH IS AVAILABLE - METERS
```

```
C                  - INCREMENTED AS SOLUTION IS MARCHED OUT IN RANGE.
C       *** RA1    - RANGE AT WHICH SOLUTION IS KNOWN - METERS
C       *** RA2    - RANGE AT WHICH SOLUTION IS TO BE SOLVED - METERS
C                    ( RA2 = RA1 + DR )
C       *** RHO1   - DENSITY IN WATER - GM/CM**3
C       *** RHO2   - DENSITY IN SEDIMENT - GM/CM**3
C       *** RMAX   - MAXIMUM RANGE OF SOLUTION - METERS
C       *** SINE   - SIN ( THETA )
C       *** TEMP   - TEMPORARY VARIABLE
C       *** THETA  - SLOPE OF BOTTOM - RADIANS
C                    THETA = 0  -  LEVEL INTERFACE
C                    THETA > 0  -  INTERFACE SLOPES DOWN
C                    THETA < 0  -  INTERFACE SLOPES UP
C       *** U      - ARRAY - COMPLEX ACOUSTIC PRESSURE FIELD
C       *** WDR    - RANGE STEP AT WHICH SOLUTION IS WRITTEN TO OUTPUT
C                    PLOTTER FILE
C       *** XK0    - REFERENCE WAVE NUMBER
C       *** XLI    - MATRIX ELEMENT, X MATRIX, LOWER DIAGONAL, ON INTERFACE
C       *** XLIZ   - XLI FOR SLOPING BOTTOM
C       *** XLAMDA - REFERENCE WAVELENGTH - METERS
C       *** XLRWS  - MATRIX ELEMENT, X MATRIX, OFF-DIAGONAL, IN WATER AND
C                    SEDIMENT
C       *** XMI    - MATRIX ELEMENT, X MATRIX, MAIN DIAGONAL, ON INTERFACE
C       *** XMS    - MATRIX ELEMENT, X MATRIX, MAIN DIAGONAL, IN SEDIMENT
C       *** XMW    - ARRAY - MATRIX ELEMENT, X MATRIX, MAIN DIAGONAL, IN
C                    WATER
C       *** XN     - REAL INDEX OF REFRACTION
C       *** XN1    - COMPLEX INDEX OF REFRACTION SQUARED
C       *** XPR    - RANGE AT WHICH SOLUTION IS PRINTED - METERS
C       *** XWR    - RANGE AT WHICH SOLUTION IS WRITTEN TO OUTPUT
C                    PLOTTER FILE
C       *** XX...  - VARIABLES THAT BEGIN WITH XX HAVE NO SPECIAL PHYSICAL
C                    SIGNIFICANCE BUT THEY CONTRIBUTE TO COMPUTATIONAL
C                    EFFICIENCY.  ALL XX VARIABLES ARE CALCULATED IN
C                    SUBROUTINE INITAL, ALL ARE INDEPENDENT OF RANGE STEP
C                    AND INTERFACE SLOPE, AND ALL ARE USED TO CALCULATE
C                    MATRIX ELEMENTS.
C       *** YLI    - MATRIX ELEMENT, Y MATRIX, LOWER DIAGONAL, ON INTERFACE
C       *** YLIV   - YLI FOR LEVEL INTERFACE
C       *** YLIZ   - YLI FOR SLOPING INTERFACE
C       *** YLRWS  - MATRIX ELEMENT, Y MATRIX, OFF-DIAGONAL, IN WATER AND
C                    SEDIMENT
C       *** YMI    - MATRIX ELEMENT, Y MATRIX, MAIN DIAGONAL, ON INTERFACE
C       *** YMS    - MATRIX ELEMENT, Y MATRIX, MAIN DIAGONAL, IN SEDIMENT
C       *** YMW    - ARRAY - MATRIX ELEMENTS, Y MATRIX, MAIN DIAGONAL, IN
C                    WATER
C       *** YRI    - MATRIX ELEMENT, Y MATRIX, UPPER DIAGONAL, ON INTERFACE
C       *** YRIV   - YRI FOR LEVEL INTERFACE
```

```
C     *** YRIZ   - YRI FOR SLOPING INTERFACE
C     *** Z1     - DEPTH OF WATER AT RANGE R1 - METERS
C     *** Z2     - DEPTH OF WATER AT RANGE R2 - METERS
C     *** ZAELYR- DEPTH OF UPPER EDGE OF ARTIFICIAL ATTENUATION
C                 LAYER - METERS
C     *** ZI     - DEPTH OF GRID POINT - METERS
C     *** ZLYR1  - MAXIMUM WATER DEPTH - METERS
C     *** ZLYR2  - DEPTH OF PRESSURE RELEASE SURFACE - METERS
C     *** ZR     - RECEIVER DEPTH - METERS
C     *** ZS     - SOURCE DEPTH - METERS
C     *** ZSVP   - ARRAY - DEPTH FOR SOUND SPEED PROFILE - METERS
C     *** ZZ...  - VARIABLES THAT BEGIN WITH ZZ HAVE NO SPECIAL PHYSICAL
C                 SIGNIFICANCE BUT THEY CONTRIBUTE TO COMPUTATIONAL
C                 EFFICIENCY.  ALL ZZ VARIABLES ARE CALCULATED IN
C                 SUBROUTINE NEWMAT, ALL DEPEND ON RANGE STEP AND/OR
C                 INTERFACE SLOPE, AND ALL ARE USED TO CALCULATE MATRIX
C                 ELEMENTS.
C     *****************************************************************
C     *** INPUT
C     *****************************************************************
C     INPUT UNIT NUMBER = NIU
C     INPUT FILENAME AND FILETYPE = IFDIN DATAIN
C     CONTENTS:  CARD IMAGES IN FREE FORMAT
C        CARD 1     :  FRQ, ZS, ZR, C0, N
C        CARD 2     :  RMAX, DRLVL, DRMAX, WDR, PDR, PDZ, PRTIN
C        CARD 3     :  BR(1), BZ(1)        **
C        CARD 4     :  BR(2), BZ(2)          *    BOTTOM PROFILE
C          .              .                 **    RANGE, WATER DEPTH (METERS)
C          .              .                  *
C        CARD N     :     .                **
C        CARD N+1   :  -1, -1
C        CARD N+2   :  ZLYR1, RHO1, BETA1
C        CARD N+3   :  ZSVP(1),CSVP(1)       **
C        CARD N+4   :  ZSVP(2),CSVP(2)         *    SOUND SPEED PROFILE
C          .              .       .           **    DEPTH (METERS) AND SOUND
C          .              .       .            *    SPEED (METERS/SEC)
C        CARD N+M   :  ZSVP(J),CSVP(J)       **
C        CARD N+M+1:  ZLYR2, RHO2, BETA2, C2
C        CARD N+M+2:  ZAELYR
C
C           WHERE:
C              FRQ     = FREQUENCY (HZ)
C              ZS      = SOURCE DEPTH (M)
C              ZR      = RECEIVER DEPTH (M)
C              C0      = REFERENCE SOUND SPEED (M/S). IF C0 = 0.0, C0 IS
C                        SET TO AVERAGE SOUND SPEED IN WATER COLUMN.
C              N       = NUMBER OF GRID POINTS
C
```

```
C          RMAX   = MAXIMUM RANGE (M) OF SOLUTION
C          DRIVL  = RANGE STEP (M) ALONG LEVEL INTERFACE. IF
C                   DRLVL = 0.0, DRLVL SEY TO 1/2 WAVELENGTH. IF
C                   DRLVL GREATER THAN DRMAX, DRIVL SET TO DRMAX.
C          DRMAX  = MAXIMUM ALLOWABLE RANGE STEP (M). IF DRMAX = 0.0,
C                   DRMAX IS SET TO ONE WAVELENGTH.
C          WDR    = RANGE STEP (M) AT WHICH SOLUTION IS  WRITTEN TO
C                   FILE USED BY PLOTTING ROUTINE.
C                   ROUNDED TO NEAREST DR.
C          PDR    = RANGE STEP (M) AT WHICH SOLUTION IS PRINTED.
C                   ROUNDED TO NEAREST DR.
C          PDZ    = DEPTH INCREMENT AT WHICH SOLUTION IS PRINTED.
C                   ROUNDED TO NEAREST DZ.
C          PRTIN  = INITIAL RANGE TO BE PRINTED AND PLOTTED (M)
C
C          BR     = RANGE (M) OF BOTTOM PROFILE
C          BZ     = DEPTH (M) OF BOTTOM PROFILE
C
C          ZLYR1  = MAXIMUM WATER DEPTH (M)
C          RHO1   = DENSITY IN WATER (GM/CM**3)
C          BETA1  = ATTENUATION (DB/WAVELENGTH) IN WATER.  IF BETA1
C                   NEGATIVE THEN ATTENUATION COMPUTED.
C
C          ZSVP   = DEPTH (M) ARRAY FOR SOUND SPEED PROFILE
C          CSVP   = SOUND SPEED (M/S) ARRAY FOR SOUND SPEED PROFILE
C
C          ZLYR2  = DEPTH (M) OF PRESSURE RELEASE SURFACE
C          RHO2   = DENSITY (GM/CM**3) IN SEDIMENT
C          BETA2  = ATTENUATION (DB/WAVELENGTH) IN SEDIMENT
C          C2     = SOUND SPEED (M/S) IN SEDIMENT
C
C          ZABLYR = DEPTH (M) OF UPPER SURFACE OF ARTIFICIAL
C                   ATTENUATION LAYER
C
C    *****************************************************************
C    *** OUTPUT
C    *****************************************************************
C       OUTPUT UNIT NUMBER FOR FILE USED BY PLOTTING ROUTINE = NOU
C       FILENAME AND FILETYPE FOR OUTPUT PLOTTER FILE = IFDOUT PLOTTER
C
C       OUTPUT UNIT NUMBER FOR PRINT FILE = NPOUT
C       FILENAME AND FILETYPE FOR OUTPUT PRINTER FILE = IFDOUT PRINTER
C    *****************************************************************
C
      COMPLEX A,A2,C,CR,CTWO,EYE,
     *      XLI,XLIZ,XLRWS,XMI,XMS,XRI,XRIZ,
     *      XX1,XX2,XX3,XX5,XX6,XX7,XX8,XX9,XX12,XX1M,
     *      YLI,YLIV,YLIZ,YLRWS,YMI,YMS,YMW,YRI,YRIV,YRIZ,
```

```
     *        U,ZZ5,ZZ6,ZZ7,ZZ8,ZZ9,ZZ10
      COMMON /IN/ IA,IBOT1,IFACE,IPZ,ISLOPE,ISTEP,IWZ,N,NA,NBOT,NM1,
     *        NSTEP,NSTEP1,NSVP,NWMAX,NXLFS
      COMMON /REAL/ ALPHA,ATT(5000),BETA1,BETA2,BR(101),BZ(101),C0,
     *        CSVP(101),C2,CWATER(5000),DR,DRLVL,DRMAX,DZ,FRQ,PDR,PDZ,
     *        R1,RA1,RA2,RHO1,RHO2,RMAX,THETA,XK0,XLAMDA,XPR,XX4,XX10,
     *        XX11,XWR,WDR,ZLYR1,ZLYR2,ZR,ZS,ZSVP(101),ZABLYR,
     *        PRTIN,PRTOT
      COMMON /CPIX/  A(5000),A2,C(5000,4),CR(5000),CTWO(5000),
     *        EYE,XLI,XLIZ,XLRWS,XMI,XMS,XRI,XRIZ,
     *        XX1,XX2,XX3,XX5,XX6,XX7,XX8,XX9,XX12,XX1M(5000),
     *        YLI,YLIV,YLIZ,YLRWS,YMI,YMS,YMW(5000),YRI,YRIV,YRIZ,
     *        U(5000),ZZ5,ZZ6,ZZ7,ZZ8,ZZ9,ZZ10
C
C     *** THE FOLLOWING CALL IS REQUIRED FOR THE NPS IBM COMPUTER.
C     *** IT PERMITS CONTINUED EXECUTION FOR ANY NUMBER OF UNDERFLOW
C     *** "ERRORS"
          CALL ERRSET (208,300,-1,1,1)
C
      CALL READ
      CALL SVPW
      CALL INITAL
      CALL MATCON
      CALL SFIELD (FRQ,C0,ZS,N,DZ,U)
      CALL WRITE1
      CALL PRTWT1
   10 CALL NEWSEG
      CALL NEWMAT
      IF ( NSTEP.EQ.1 ) GO TO 10
C
C     *** THIS LOOP ADVANCES THE SOLUTION ALONG ONE ENTIRE LINEAR
C     *** BOTTOM SEGMENT
          DO 80 ISTEP=2,NSTEP
C             *** UPDATE RANGES
                  RA1 =  R1 + (ISTEP-1)*DR
                  RA2 =  RA1 + DR
C             *** IS SEGMENT SLOPING DOWN, LEVEL, UP, MODIFIED DOWN,
C             *** OR MODIFIED UP?
                  GO TO (20,30,40,50,60), ISLOPE
C                 *** BOTTOM SLOPES DOWN
   20                 CALL DOWN
                      GO TO 70
C                 *** BOTTOM IS LEVEL
   30                 CALL LEVEL
                      GO TO 70
C                 *** BOTTOM SLOPES UP
   40                 CALL UP
                      GO TO 70
```

```
C                 *** BOTTOM SLOPES DOWN SLOWLY, BOTTOM MODIFIED
 50                   CALL SSLOPE
                      GO TO 70
C                 *** BOTTOM SLOPES UP SLOWLY, BOTTOM MODIFIED
 60                   CALL SSLOPE
C
 70             CONTINUE
                CALL ATTENU(U,ATT,IA,NA)
                RA2P = RA2+0.5
C               *** TIME TO WRITE?
                    IF (RA2P.GE.XWR) CALL WRITE2
C               *** TIME TO PRINT?
                    IF (RA2P.GE.XPR) CALL PRTWT2
C               *** TIME TO TERMINATE?
                    IF (RA2.GE.RMAX) GO TO 90
 80           CONTINUE
C
C       *** GO BACK AND CONTINUE WITH NEXT LINEAR BOTTOM SEGMENT
              GO TO 10
C
C       *** TIME TO TERMINATE
  90    CONTINUE
C           CALL END (RA2)
        STOP
        END
C
C
C
C
        SUBROUTINE READ
C
C          (1)   THIS SUBROUTINE READS ALL INPUT DATA.
C          (2)   THE DATA IS READ FROM INPUT UNIT NUMBER:  NIU = 51
C          (3)   INPUT FILENAME AND FILETYPE ARE:  IFDIN  DATAIN
C          (4)   DATA IS READ IN FREE FORMAT.
C          (5)   DATA IS TRANSFERRED BACK TO MAIN PROGRAM VIA COMMON BLOCK
C
        COMMON /IN/ IA,IBOT1,IFACE,IPZ,ISLOPE,ISTEP,IWZ,N,NA,NBOT,NM1,
     *          NSTEP,NSTEP1,NSVP,NWMAX,NXLFS
        COMMON /REAL/ ALPHA,ATT(5000),BETA1,BETA2,BR(101),BZ(101),C0,
     *          CSVP(101),C2,CWATER(5000),DR,DRLVL,DRMAX,DZ,FRQ,PDR,PDZ,
     *          R1,RA1,RA2,RHO1,RHO2,RMAX,THETA,XK0,XLAMDA,XPR,XX4,XX10,
     *          XX11,XWR,WDR,ZLYR1,ZLYR2,ZR,ZS,ZSVP(101),ZABLYR,
     *          PRTIN,PRTOT
CCCC    DATA NIU/5/, NPOUT/6/
        DATA NIU/51/, NPOUT/55/
C
C       *** READ INPUT PARAMETERS
```

```
          READ(NIU,*,END=100) FRQ, ZS, ZR, CO, N
          READ(NIU,*,END=100) RMAX, DRLVL, DRMAX, WDR, PDR, PDZ, PRTIN
          IF(PRTIN .EQ. 0.0) PRTIN = WDR
          PRTOT = RMAX
          IF(PRTOT .LE. PRTIN) GO TO 90
C
C     *** READ BOTTOM PROFILE - RANGE, DEPTH
          DO 10 I=1,101
             READ(NIU,*,END=100) BR(I), BZ(I)
             NBOT=I
C            *** END OF PROFILE?
                 IF(BR(I).LT.0.0) GO TO 20
C            *** NO
 10       CONTINUE
C
 20   CONTINUE
C     *** EXTEND LAST DEPTH BEYOND MAX RANGE
          BR(NBOT) = 1.0E+10
          BZ(NBOT) = BZ(NBOT-1)
C
C     *** FIRST LAYER IS WATER. SECOND IS SEDIMENT.
C     *** READ MAX DEPTH, DENSITY AND ATTENUATION OF FIRST LAYER
          READ(NIU,*,END=100) ZLYR1, RHO1, BETA1
C
C     *** READ SOUND SPEED PROFILE IN FIRST LAYER
          DO 25 I=1,101
             NSVP=I
             READ(NIU,*,END=100) ZSVP(I), CSVP(I)
C            *** READ ANOTHER PROFILE POINT?
                 IF(ZSVP(I).LT.ZLYR1) GO TO 25
C            *** NO
C            *** WAS THAT THE LAST PROFILE POINT?
                 IF(ZSVP(I).EQ.ZLYR1) GO TO 30
C            *** NO, THERE IS ERROR.
                 GO TO 101
 25       CONTINUE
C
C     *** DOES THE SOUND SPEED PROFILE START AT THE SURFACE?
 30       IF ( ZSVP(1).NE.0.0 ) GO TO 102
C     *** YES
C
C     *** READ DEPTH, DENSITY, ATTENUATION AND SPEED IN SECOND LAYER
          READ(NIU,*,END=100) ZLYR2, RHO2, BETA2,C2
C     *** READ DEPTH OF UPPER EDGE OF ARTIFICIAL ATTENUATING LAYER
          READ(NIU,*,END=100) ZABLYR
      RETURN
C
C     *** ERROR EXITS
```

```
 90   WRITE(6,899)
      WRITE(NPOUT,899)
      STOP
 100  WRITE(6,900)
      WRITE(NPOUT,900)
      STOP
 101  WRITE(6,901)
      WRITE(NPOUT,901)
      STOP
 102  WRITE(6,902)
      WRITE(NPOUT,902)
      STOP
C
 899  FORMAT(//,1X,'ERROR: THE INITIAL PRINTER/PLOTTER RANGE IS',/,
     *      9X,'GREATER THAN  OR EQUAL TO THE FINAL RANGE',/,
     *      9X,'EXECUTION TERMINATED',//)
 900  FORMAT(//,1X,'ERROR:  EXPECTING MORE INPUT DATA. ',/, 9X,
     *      'EXECUTION TERMINATED.',//)
 901  FORMAT(//,' ERROR:  FINAL DEPTH IN SOUND SPEED PROFILE DOES NOT '
     *      ,/,9X,'EQUAL MAXIMUM DEPTH OF WATER COLUMN.',//,9X,
     *      'EXECUTION TERMINATED.',//)
 902  FORMAT(//,' ERROR:  FIRST DEPTH IN SOUND SPEED PROFILE'
     *      ,/,9X,'DOES NOT EQUAL ZERO.',//,9X,
     *      'EXECUTION TERMINATED.',//)
C
      END
C
C
C
C
      SUBROUTINE SVPW
C
C        (1)  THIS SUBROUTINE CALCULATES THE VERTICAL STEP SIZE:  DZ
C        (2)  THIS SUBROUTINE ALSO CALCULATES THE SPEED OF SOUND AT
C             EACH OF THE VERTICAL GRID POINTS.
C        (3)  SOUND SPEED VALUES ARE DETERMINED BY LINEAR INTERPOLATION.
C        (4)  SOUND SPEEDS ARE STORED IN CWATER(I).
C             (A)  THE INDEX I RANGES FROM 1 TO NWMAX.
C             (B)  CWATER(1) CORRESPONDS TO THE GRID POINT DZ BELOW
C                  THE SURFACE.
C
      COMMON /IN/ IA,IBOT1,IFACE,IPZ,ISLOPE,ISTEP,IWZ,N,NA,NBCT,NM1,
     *        NSTEP,NSTEP1,NSVP,NWMAX,NXLFS
      COMMON /REAL/ ALPHA,ATT(5000),BETA1,BETA2,BR(101),BZ(101),C0,
     *        CSVP(101),C2,CWATER(5000),DR,DRLVL,DRMAX,DZ,FRQ,PDR,PDZ,
     *        R1,RA1,RA2,RHO1,RHO2,RMAX,THETA,XK0,XLAMDA,XPR,XX4,XX10,
     *        XX11,XWR,WDR,ZLYR1,ZLYR2,ZR,ZS,ZSVP(101),ZABLYR,
     *        PRTIN,PRTOT
```

```
C
C     ***CALCULATE VERTICAL STEP SIZE
         DZ = ZLYR2 / FLOAT(N)
C
C     ***CALCULATE NUMBER OF GRID POINTS IN WATER COLUMN
         NWMAX = INT((ZLYR1/DZ)+0.5)
C
C     ***CALCULATE SOUND SPEED AT ALL GRID POINTS IN WATER COLUMN
         L = 1
         DO 20 I=1,NWMAX
            ZI = I*DZ
            LP1 = L+1
C           ***NEED TO UPDATE PROFILE ENDPOINTS?
               IF (ZI.LE.ZSVP(LP1)) GO TO 10
C           ***YES
               L = L+1
               LP1 = L+1
 10         CWATER(I) = CSVP(L) + (CSVP(LP1)-CSVP(L)) * (ZI-ZSVP(L)) /
     *                  (ZSVP(LP1)-ZSVP(L))
 20      CONTINUE
C
      RETURN
      END
C
C
C
C
      SUBROUTINE INITAL
C
C        (1)  THIS SUBROUTINE INITIALIZES CONSTANTS AND VARIABLES.
C        (2)  VALUES ARE TRANSFERRED TO/FROM MAIN PROGRAM VIA COMMON
C             BLOCK.
C
      COMMON /IN/ IA,IBOT1,IFACE,IPZ,ISLOPE,ISTEP,IWZ,N,NA,NBOT,NM1,
     *        NSTEP,NSTEP1,NSVP,NWMAX,NXLFS
      COMMON /REAL/ ALPHA,ATT(5000),BETA1,BETA2,BR(101),BZ(101),C0,
     *        CSVP(101),C2,CWATER(5000),DR,DRLVL,DRMAX,DZ,FRQ,PDR,PDZ,
     *        R1,RA1,RA2,RHO1,RHO2,RMAX,THETA,XK0,XLAMDA,XPR,XX4,XX10,
     *        XX11,XWR,WDR,ZLYR1,ZLYR2,ZR,ZS,ZSVP(101),ZABLYR,
     *        PRTIN,PRTOT
      DATA PI/3.141592654/
CCC   DATA PI/3.14/
C
C     *** IF C0 NOT SPECIFIED, SET C0 TO AVERAGE SPEED IN WATER COLUMN
C     *** (USING MAX DEPTH PROFILE)
         IF(C0.NE.0.0) GO TO 11
         DO 10 I=2,NSVP
            C0=C0+(ZSVP(I)-ZSVP(I-1))*(CSVP(I-1) +
```

```
     *              0.5  *(CSVP(I)-CSVP(I-1)))
 10        CONTINUE
           CO = CO/ZSVP(NSVP)
 11        CONTINUE
C
C     *** INITIALIZE RANGE
           RA1 = 0.0
C
C     *** INITIALIZE POINTER THAT POINTS TO BOTTOM PROFILE POINT
           IBCT1 = 0
C
C     *** COMPUTE REFERENCE WAVE NUMBER
           XKO = 2.0*PI*FRQ/CO
C
C     *** COMPUTE REFERENCE WAVELENGTH
           XLAMDA = CO/FRQ
C
C     *** IF DRLVL=0 SET DRLVL EQUAL TO 1/2 REFERENCE WAVELENGTH
           IF ( DRLVL.EQ.0.0 )   DRLVL = 0.5 * XLAMDA
C
C     *** IF DRMAX=0 SET DRMAX EQUAL TO REFERENCE WAVELENGTH
           IF ( DRMAX.EQ.0.0 ) DRMAX = XLAMDA
C
C     *** IF DRLVL GREATER THAN DRMAX SET DRLVL EQUAL TO DRMAX
           IF (DRLVL.GT.DRMAX) DRLVL = DRMAX
C
C     *** COMPUTE ATTENUATION - SACLANT MEMO SM-121 (JENSEN + FERLA)
C     *** MODIFIED AS FOLLOWS:
C     *** IF INPUTTED BETA IS LT 0.0, ALPHA IS COMPUTED IN DB/METER
C     *** AND USED FOR BETA
           ALPHA=FRQ*FRQ*(.007+(.155*1.7)/(1.7*1.7+FRQ*FRQ*.000001))
     *          *1.0E-09
C
C     *** INITIALIZE POINTER THAT POINTS TO INTERFACE GRID POINT
           IFACE = INT ( BZ(1)/DZ + 0.5 )
C
        RETURN
        END
C
C
C
C
        SUBROUTINE MATCON
C
C       THIS SUBROUTINE CALCULATES VARIOUS VARIABLES NEEDED TO COMPUTE
C       TRIDIAGINAL MATRIX ELEMENTS.  VARIABLES BEGINNING WITH XX HAVE
C       NO SPECIAL PHYSICAL SIGNIFICANCE BUT THEY CONTRIBUTE TO
C       COMPUTATIONAL EFFICIENCY.
```

```
C
      CCMPLEX XN1
      COMPLEX A,A2,C,CR,CTWO,EYE,
     *        XLI,XLIZ,XLRWS,XMI,XMS,XRI,XRIZ,
     *        XX1,XX2,XX3,XX5,XX6,XX7,XX8,XX9,XX12,XX1M,
     *        YLI,YLIV,YLIZ,YLRWS,YMI,YMS,YMW,YRI,YRIV,YRIZ,
     *        U,ZZ5,ZZ6,ZZ7,ZZ8,ZZ9,ZZ10
      COMMON /IN/ IA,IBOT1,IFACE,IPZ,ISLOPE,ISTEP,IWZ,N,NA,NBOT,NM1,
     *        NSTEP,NSTEP1,NSVP,NWMAX,NXLFS
      CCMMON /REAL/ ALPHA,ATT(5000),BETA1,BETA2,BR(101),BZ(101),C0,
     *        CSVP(101),C2,CWATER(5000),DR,DRLVL,DRMAX,DZ,FRQ,PDR,PDZ,
     *        R1,RA1,RA2,RHO1,RHO2,RMAX,THETA,XK0,XLAMDA,XPR,XX4,XX10,
     *        XX11,XWR,WDR,ZLYR1,ZLYR2,ZR,ZS,ZSVP(101),ZABLYR,
     *        PRTIN,PRTOT
      COMMON /CPLX/  A(5000),A2,C(5000,4),CR(5000),CTWO(5000),
     *        EYE,XLI,XLIZ,XLRWS,XMI,XMS,XRI,XRIZ,
     *        XX1,XX2,XX3,XX5,XX6,XX7,XX8,XX9,XX12,XX1M(5000),
     *        YLI,YLIV,YLIZ,YLRWS,YMI,YMS,YMW(5000),YRI,YRIV,YRIZ,
     *        U(5000),ZZ5,ZZ6,ZZ7,ZZ8,ZZ9,ZZ10
C
      EYE = CMPLX(0.0,1.0)
C
      XX1 = CMPLX ( 0.0 , +0.25/(DZ*DZ*XK0) )
      XX2 = 2.0 * XX1
C     *** COMPUTE COEFFICIENT A IN SEDIMENT LAYER
C        *** FIRST CALCULATE REAL INDEX OF REFRACTICN
            XN = C0/C2
C        *** THEN CALCULATE COMPLEX INDEX OF REFRACTICN SQUARED
C        ***     (SEE PAGE 2-11 IN TR 6659)
            XN1 = CMPLX ( XN*XN , XN*XN*BETA2/27.287527 )
C        *** CALCULATE A2
            A2 = 0.5 * EYE * XK0 * (XN1-1.0)
      XX3 = 0.5*A2 - XX2
      XX4 = 1.0 + RHO1/RHO2
      XX5 = XX2*(1.0/XX4 - 1.0) + A2/(2.0*(RHO2/RHO1 + 1.0))
      XX6 = XX2/XX4
      XX7 = RHO1/RHO2 * XX6
      XX8 = CMPLX ( 0.0 , -DZ*XK0 )
      XX9 = 1.0 / XX8
      XX10 = RHO1 + RHO2
      XX11 = RHO1 - RHO2
      XX12 = 4.0 * XX1
C
C        *** THIS SECTION PERTAINS TO POINTS IN WATER COLUMN
           DO 10 I=1,NWMAX
C              *** CALCULATE REAL INDEX OF REFRACTICN IN WATER
                  XN = C0/CWATER(I)
C              *** CALCULATE ATTENUATION AS PER CCMMENTS IN SUBROUTINE
```

```
C             *** INITAL
                  IF (BETA1.LT.0.0) BETA1 = ALPHA*CWATER(I)/FRQ
C             *** CALCULATE COMPLEX INDEX OF REFRACTION SQUARED
C             ***   (SEE PAGE 2-11 IN TR 6659)
                  XN1 = CMPLX ( XN*XN , XN*XN*BETA1/27.287527 )
C             *** CALCULATE COEFFICIENT A(I)
                  A(I) = 0.5 * EYE * XKO * (XN1-1.0)
C             *** CALCULATE XX1M
                  XX1M(I) = 0.5 * A(I)  -  XX2
 10         CONTINUE
      RETURN
      END
C
C
C
C
      SUBROUTINE SFIELD(FRQ,CO,ZS,N,DZ,U)
C
C     *** THIS SUBROUTINE IS IDENTICAL TO SUBROUTINE SFIELD AS PER
C     *** NUSC TECHNICAL REPORT 6659.
C
C     ****************************************************************
C     *** GAUSSIAN STARTING FIELD - SEE NORDA TECH NOTE 12 BY H.K.BROCK
C     ****************************************************************
C
C     *** CALLING ROUTINE SUPPLIES:
C         FRQ - FREQUENCY IN HZ
C         CO  - REFERENCE SOUND SPEED - METERS/SEC
C         ZS  - DEPTH OF SOURCE IN METERS.
C         N   - NUMBER OF POINTS IN ARRAY U
C         DZ  - DEPTH INCREMENT - METERS
C     *** SFIELD SUBROUTINE SUPPLIES:
C         U   - COMPLEX STARTING FIELD
C     ****************************************************************
C
      COMPLEX U(1)
      DATA PI/3.1415926535/
CCC   DATA PI/3.14/
C
C     THE FIELD IS DEFINED AS A GAUSSIAN BEAM AT RANGE = 0.
C     LOCAL VARIABLES - GA  GAUSSIAN AMPLITUDE
      XKO=2.0*PI*FRQ/CO
      GW=2.0/XKO
      GA=SQRT(GW)/GW
      DO 10 I=1,N
      ZM=I*DZ
      PR=GAUSS(GA,ZM,ZS,GW)-GAUSS(GA,-ZM,ZS,GW)
      U(I)=CMPLX(PR,0.0)
```

```
 10     CONTINUE
        RETURN
        END
        FUNCTION GAUSS(GA,Z,GD,GW)
 C      INPUT  - GA  GAUSSIAN AMPLITUDE
 C      OUTPUT - GAUSS = GA * EXP(-((Z - GD) / GW)**2)
 C      TEMPORARY VARIABLE - V
        V=(Z-GD)/GW
        V=-(V*V)
        GAUSS=GA*EXP(V)
        RETURN
        END
 C
 C
 C
 C
        SUBROUTINE WRITE1
 C
 C         (1)  THIS SUBROUTINE OUTPUTS UNFORMATTED DATA TO A FILE
 C              THAT IS USED BY THE PLOTTING ROUTINE.
 C         (2)  THE FILE CORRESPONDS TO UNIT FILE NUMBER:  NOU = 52
 C         (3)  THE FILENAME AND FILETYPE FOR THIS FILE ARE:
 C                   IFDOUT   PLOTTER
 C
        COMPLEX A,A2,C,CR,CTWO,EYE,
      *         XLI,XLIZ,XLRWS,XMI,XMS,XRI,XRIZ,
      *         XX1,XX2,XX3,XX5,XX6,XX7,XX8,XX9,XX12,XX1M,
      *         YLI,YLIV,YLIZ,YLRWS,YMI,YMS,YMW,YRI,YRIV,YRIZ,
      *         U,ZZ5,ZZ6,ZZ7,ZZ8,ZZ9,ZZ10
        COMMON /IN/ IA,IBOT1,IFACE,IPZ,ISLOPE,ISTEP,IWZ,N,NA,NBCT,NM1,
      *         NSTEP,NSTEP1,NSVP,NWMAX,NXLFS
        COMMON /REAL/ ALPHA,ATT(5000),BETA1,BETA2,BR(101),BZ(101),CO,
      *         CSVP(101),C2,CWATER(5000),DR,DRLVL,DRMAX,DZ,FRQ,PDR,PDZ,
      *         R1,RA1,RA2,RHO1,RHO2,RMAX,THETA,XKO,XLAMDA,XPR,XX4,XX10,
      *         XX11,XWR,WDR,ZLYR1,ZLYR2,ZR,ZS,ZSVP(101),ZABLYR,
      *         PRTIN,PRTOT
        COMMON /CPLX/  A(5000),A2,C(5000,4),CR(5000),CTWO(5000),
      *         EYE,XLI,XLIZ,XLRWS,XMI,XMS,XRI,XRIZ,
      *         XX1,XX2,XX3,XX5,XX6,XX7,XX8,XX9,XX12,XX1M(5000),
      *         YLI,YLIV,YLIZ,YLRWS,YMI,YMS,YMW(5000),YRI,YRIV,YRIZ,
      *         U(5000),ZZ5,ZZ6,ZZ7,ZZ8,ZZ9,ZZ10
        DATA NOU/52/
 C
 C      *** WRITE INITIAL AND FINAL RANGE FOR THE X-AXIS OF PLOT
            WRITE(NOU,*) PRTIN,PRTOT
 C
 C      *** INITIALIZE RANGE VARIABLE AT WHICH SOLUTION IS TO BE RECORDED
            XWF = RA1+WDR
```

```
C
C     *** COMPUTE RECEIVER DEPTH TO NEAREST DZ
          IF ( ZR.LT.DZ ) ZR = DZ
          IWZ = ZR/DZ + 0.5
          ZR = IWZ*DZ
C
C     *** WRITE STARTING VALUE
          IF((RA1 .LT. PRTIN) .OR. (RA1 .GT. PRTOT)) GO TO 100
          WRITE(NCU,*)  RA1, ZR, U(IWZ)
C
  100 CCNTINUE
      RETURN
      END
C
C
C
C
      SUBROUTINE PRTWT1
C
C     THIS SUBROUTINE IS USED TO OUTPUT DATA IN A FCRMAT WHICH IS
C     REDUCED IN VOLUME AND COMPATIBLE FOR THE WAVENUMBER TECHNIQUE
C     (WT)
C
C
C        (1)  THIS SUBROUTINE OUTPUTS FORMATTED DATA TO A FILE
C             WHICH IS READY TO BE SENT TC THE PRINTER.
C        (2)  THE FILE CORRESPONDS TO UNIT FILE NUMBER:  NPOUT = 55
C        (3)  THE FILENAME AND FILETYPE FOR THIS FIIE ARE:
C                 IFDOUT     PRINTER
C
      DIMENSION ID(17)
      CCMMON /IN/ IA,IBOT1,IFACE,IPZ,ISLOPE,ISTEP,IWZ,N,NA,NBCT,NM1,
     *        NSTEP,NSTEP1,NSVP,NWMAX,NXLFS
      CCMMON /REAL/ ALPHA,ATT(5000),BETA1,BETA2,BR(101),BZ(101),CO,
     *        CSVP(101),C2,CWATER(5000),DR,DRLVL,DRMAX,DZ,FRQ,PDR,PDZ,
     *        R1,RA1,RA2,RHO1,RHO2,RMAX,THETA,XK0,XLAMDA,XPR,XX4,XX10,
     *        XX11,XWR,WDR,ZLYR1,ZLYR2,ZR,ZS,ZSVP(101),ZABLYR,
     *        PRTIN,PRTOT
      DATA NPOUT/55/
C
C     *** PRCMPT USER FOR RUN IDENTIFICATION
          WRITE(6,890)
  890 FCRMAT(T4,'INPUT TITLE...MAX OF 60 CHARACTERS')
C     *** READ USER RESPONSE
          READ(5,891) ( ID(I), I=1,15 )
  891 FCRMAT(15A4)
C
C     *** PRINT SELECTED PARAMETERS OF INTEREST
```

```
            WRITE(NFOUT,900) (ID(I), I=1,15),FRQ,ZS,ZR,CO,RMAX,ZLYR1,
     *      BETA1,XKO,XLAMDA,DZ,WDR,ZABLYR,N
 900  FORMAT(T5,15A4,/,8(1X,F9.3),/,4(1X,F9.3),1X,I10)
      IPZ = INT(PDZ/(DZ+0.5))
      IF(IPZ .EQ. 0) IPZ = 1
      XPR = RA1 + PDR
C
      RETURN
      END
C
C
C
C
      SUBROUTINE NEWSEG
C
C     THIS SUBROUTINE IS CALLED AT THE START OF EACH NEW BOTTOM
C     SEGMENT.  THE SUBROUTINE DOES THE FOLLOWING TASKS FOR EACH
C     BOTTOM SEGMENT:
C        (1)  UPDATES BOTTOM PROFILE POINTERS:  IBOT1 & IBOT2
C        (2)  COMPUTES SLOPE:  THETA
C        (3)  COMPUTES NUMBER OF RANGE STEPS IN SEGMENT:  NSTEP
C        (4)  COMPUTES RANGE STEP:  DR
C        (5)  SETS SLOPE FLAG :  ISLOPE
C              (A)  ISLOPE = 1   -   BOTTOM SLOPES DOWN
C              (B)  ISLOPE = 2   -   BOTTOM LEVEL
C              (C)  ISLOPE = 3   -   BOTTOM SLOPES UP
C              (D)  ISLOPE = 4   -   BOTTOM SLOPES DOWN, MODIFY BOTTOM
C              (E)  ISLOPE = 5   -   BOTTOM SLOPES UP, MODIFY BOTTOM
C        (6)  INIALIZES RANGES:  RA1 & RA2
C        (7)  CHECKS THAT RANGE STEP IS LESS THAN DRMAX
C        (8)  ISSUES ERROR OR WARNING MESSAGES AS APPROPRIATE
C
      COMMON /IN/ IA,IBOT1,IFACE,IPZ,ISLOPE,ISTEP,IWZ,N,NA,NBOT,NM1,
     *         NSTEP,NSTEP1,NSVP,NWMAX,NXLFS
      COMMON /REAL/ ALPHA,ATT(5000),BETA1,BETA2,BR(101),BZ(101),CO,
     *         CSVP(101),C2,CWATER(5000),DR,DRLVL,DRMAX,DZ,FRQ,PDR,PDZ,
     *         R1,RA1,RA2,RHO1,RHO2,RMAX,THETA,XKO,XLAMDA,XPR,XX4,XX10,
     *         XX11,XWR,WDR,ZLYR1,ZLYR2,ZR,ZS,ZSVP(101),ZABLYR,
     *         PRTIN,PRTOT
      DATA NFOUT/55/
C
C     *** UPDATE BOTTOM PROFILE POINTER
            IBOT1 = IBOT1 + 1
            IBOT2 = IBOT1 + 1
C     *** GET STARTING AND ENDING RANGES AND DEPTHS FOR THIS SEGMENT
            R1 = BR(IBOT1)
            Z1 = BZ(IBOT1)
            R2 = BR(IBOT2)
```

```
         Z2 = B2(IBOT2)
C        *** ERROR CHECK
            IF (R2.LE.R1) GO TO 100
C        *** PUT Z1 AND Z2 ON NEAREST GRID POINTS
            ITEMP = INT ( Z1/DZ + 0.5 )
            Z1 = DZ * FLOAT(ITEMP)
            ITEMP = INT ( Z2/DZ + 0.5 )
            Z2 = DZ * FLOAT(ITEMP)
C     *** COMPUTE SLOPE
         THETA = ATAN2 (Z2-Z1,R2-R1)
C     *** DOES BOTTOM SLOPE DOWN, LEVEL OR UP?
         IF (THETA.GT.0.0) GO TO 10
         IF (THETA.LT.0.0) GO TO 20
C
C        *** BOTTOM IS LEVEL
C           *** DETERMINE NUMBER OF RANGE STEPS FOR SEGMENT
               NSTEP = INT ( (R2-R1)/DRLVL + 0.99999 )
C           *** DETERMINE RANGE STEP
               DR = (R2-R1) / FLOAT(NSTEP)
C           *** SET ISLOPE
               ISLOPE = 2
           GO TO 80
C
C       *** BOTTOM SLOPES DOWN
C          *** DETERMINE NUMBER OF RANGE STEPS
 10            NSTEP = INT ( (Z2-Z1+0.05)/DZ )
C          *** DETERMINE RANGE STEP
               DR = (R2-R1)/FLOAT(NSTEP)
C          *** SET ISLOPE
               ISLOPE = 1
          GO TO 30
C
C       *** BOTTOM SLOPES UP
C          *** DETERMINE NUMBER OF RANGE STEPS
 20            NSTEP = INT ( (Z1-Z2+0.05)/DZ )
C          *** DETERMINE RANGE STEP
               DR = (R2-R1)/FLOAT(NSTEP)
C          *** SET ISLOPE
               ISLOPE = 3
C
 30   CONTINUE
C     *** IS RANGE STEP TOO LARGE?
         IF ( DR.LE.DRMAX ) GO TO 80
C     *** YES, BOTTOM MUST BE MODIFIED
C        *** SET ISLOPE
            ISLOPE = 4
            IF ( THETA.LT.0.0 ) ISLOPE = 5
C        *** DETERMINE NUMBER OF RANGE STEPS REQUIRED TO MOVE UP
```

```
C           *** OR DOWN ONE GRID POINT
                NSTEP1 = INT ( DR/DRMAX + 0.99999 )
C           *** DETERMINE RANGE STEP
                DR = DR / FLOAT(NSTEP1)
C           *** REDETERMINE NUMBER OF RANGE STEPS
                NSTEP = NSTEP * NSTEP1
C           *** COMPUTE SLOPE OF SLOPING SECTION
                THETA = ATAN2(DZ,DR)
C           *** COMPUTE LOCATION OF NEXT LEVEL SECTION FOLLOWING A
C           *** SLOPING SECTION
                NXLPS = NSTEP1/2 + 2
C           *** INDICATE TO USER THAT BOTTOM HAS BEEN MODIFIED
                TEMP = 0.5 * DZ
                WRITE(6,903) R1,R2,TEMP
                WRITE(NPOUT,903) R1,R2,TEMP
C
 80     CONTINUE
C       *** INITIALIZE RA1 & RA2
            RA1 = R1
            RA2 = RA1+DR
C
C       *** INDICATE TO USER HOW FAR SOLUTION FIELD HAS PROGRESSED
            WRITE(6,902) R1
C
C       *** IF RANGE STEP GREATER THAN 1 (?) WAVELENGTH WRITE WARNING
            IF ( DR.LE.XLAMDA ) GO TO 90
               WRITE(6,901) R1, R2, DR, XLAMDA
               WRITE(NPOUT,901) R1, R2, DR, XLAMDA
C
 90     RETURN
C
C       *** ERROR EXIT
 100        WRITE(6,900) IBOT2, IBOT1
            WRITE(NPOUT,900) IBOT2, IBOT1
C
            STOP
C
 900    FORMAT(//,1X,'ERROR:  THE RANGE AT BOTTOM PROFILE POINT NUMBER ',
     *       I2,' IS LESS'/,9X,'THAN THE RANGE AT BOTTOM PROFILE POINT'
     *       ,' NUMBER ',I2,'.',//,1X,'EXECUTION TERMINATED.'///)
 901    FORMAT(/,
     * ' WARNING:  THE HORIZONTAL RANGE STEP BETWEEN RANGE R  =',F8.1,/
     * '           AND RANGE R =',F8.1,' (METERS) IS',F5.1,' METERS.',/,
     * '           THE REFERENCE WAVELENGTH IS',F5.1,' METERS.')
 902    FORMAT(/,' THE PROGRAM HAS REACHED RANGE R =',F8.1,' METERS.')
 903    FORMAT(/,' NOTE:  THE BOTTOM BETWEEN RANGE',F8.1,' AND RANGE',
     *      F8.1,/,8X,'HAS BEEN MODIFIED BECAUSE OF ITS VERY SMALL',
     *      ' SLOPE.',/,8X,'THE DIFFERENCE BETWEEN THE MODIFIED ',
```

```
     *        'BOTTCM AND YOUR',/,8X,'INPUT BOTTCM IS NEVER GREATER ',
     *        'THAN',F5.2,' METERS.'/)
C
      END
C
C
C
C
      SUBROUTINE NEWMAT
C
C
C     THIS SUBROUTINE IS CALLED AT THE START OF EACH NEW BOTTCM
C     SEGMENT.  THE SUBROUTINE DOES THE FOLLOWING TASKS:
C        (1)  COMFUTES TRIIIAGONAL MATRIX ELEMENTS FOR MATRIX Y
C             (A)  MATRIX Y IS AT RANGE WHERE FIELD IS KNOWN: RANGE=RA1
C             (B)  Y MATRIX ON RHS OF EQUATION
C        (2)  USES Y MATRIX ELEMENTS AND KNOWN FIELD AT RANGE RA1 TC
C             COMFUTE RHS COLUMN VECTOR C(I,4).    (SEE TRIDG AND TEXT)
C        (3)  COMFUTES TRIDIAGONAL MATRIX ELEMENTS FOR MATRIX X
C             (A)  X MATRIX ON LHS OF EQUATION
C             (B)  X MATRIX VALUES STORED IN C(I,1), C(I,2) AND C(I,3)
C                  (SEE TRIDG AND TEXT REFERENCED IN TRIDG)
C             (C)  X MATRIX AT RANGE:  RA2 = RA1 + DR
C
C     TRIDIAGONAI MATRIX ELEMENTS ARE REPRESENTED BY VARIABLES
C     BEGINNING WITH EITHER X OR Y.  FOR VARIABLES BEGINNING WITH X OR
C     Y, THE LETTERS IN THE VARIABLE NAME HAVE THE FOLLOWING
C     SIGNIFICANCE:
C                     X   -   ELEMENT IN X MATRIX
C                     Y   -   ELEMENT IN Y MATRIX
C                     M   -   ELEMENT ON MAIN DIAGONAL
C                     L   -   ELEMENT TO LEFT OF MAIN DIAGONAL
C                     R   -   ELEMENT TO RIGHT OF MAIN DIAGONAL
C                     W   -   ELEMENT IN ROW IN WATER
C                     I   -   ELEMENT IN ROW AT INTERFACE
C                     S   -   ELEMENT IN ROW IN SEDIMENT
C                     V   -   INTERFACE ELEMENT ON IEVFL INTERFACE
C                     Z   -   INTERFACE ELEMENT ON SLOPING INTERFACE
C
      CCMPLEX DEITA,DELIN,BEDA1,GAMMA1,BEDA2,GAMMA2,ZZ1,ZZ2,ZZ3,ZZ4
      CCMPLEX A,A2,C,CR,CTWO,EYE,
     *        XLI,XLIZ,XLRWS,XMI,XMS,XRI,XRIZ,
     *        XX1,XX2,XX3,XX5,XX6,XX7,XX8,XX9,XX12,XX1M,
     *        YLI,YLIV,YLIZ,YLRWS,YMI,YMS,YMW,YRI,YRIV,YRIZ,
     *        U,ZZ5,ZZ6,ZZ7,ZZ8,ZZ9,ZZ10
      COMMON /IN/ IA,IBOT1,IFACE,IPZ,ISLOPE,ISTEP,IWZ,N,NA,NBOT,NM1,
     *        NSTEP,NSTEP1,NSVP,NWMAX,NXLFS
      CCMMON /REAL/ ALPHA,ATT(5000),BETA1,BETA2,BR(101),BZ(101),CO,
```

```
     *        CSVP(101),C2,CWATER(5000),DR,DRLVL,DRMAX,DZ,FRQ,PDR,PDZ,
     *        R1,RA1,RA2,RHO1,RHO2,RMAX,THETA,XK0,XLAMDA,XPR,XX4,XX10,
     *        XX11,XWR,WDR,ZLYR1,ZLYR2,ZR,ZS,ZSVP(101),ZABLYR,
     *        PRTIN,PRTOT
      COMMON /CPIX/  A(5000),A2,C(5000,4),CR(5000),CTWO(5000),
     *        EYE,XLI,XLIZ,XLRWS,XMI,XMS,XRI,XRIZ,
     *        XX1,XX2,XX3,XX5,XX6,XX7,XX8,XX9,XX12,XX1M(5000),
     *        YLI,YLIV,YLIZ,YLRWS,YMI,YMS,YMW(5000),YRI,YRIV,YRIZ,
     *        U(5000),ZZ5,ZZ6,ZZ7,ZZ8,ZZ9,ZZ10
C
C     *** COMPUTE MAIN DIAGONAL ELEMENT, Y MATRIX, IN SEDIMENT
         YMS =  1.0 + DR*XX3
C     *** COMPUTE OFF-DIAGONAL ELEMENTS, Y MATRIX, IN WATER & SEDIMENT
         YLRWS  = DR*XX1
C     *** COMPUTE MAIN DIAGONAL ELEMENT, X MATRIX, IN SEDIMENT
         XMS =  2.0 - YMS
C     *** COMPUTE OFF-DIAGONAL ELEMENTS, X MATRIX, IN WATER & SEDIMENT
         XLRWS  = -YLRWS
C     *** COMPUTE FIRST ELEMENT IN RHS COLUMN VECTOR
         YMW(1)  = 1.0 + DR*XX1M(1)
         C(1,4)  = U(1)*YMW(1)  + U(2)*YLRWS
C     *** COMPUTE TWO ELEMENTS IN FIRST ROW ON LHS
         C(1,2)  = 2.0 - YMW(1)
         C(1,3)  = XLRWS
C     *** COMPUTE REMAINING ELEMENTS ON BOTH RHS & LHS FOR ROWS IN WATER
         IFACEM = IFACE - 1
         DO 10 I=2,IFACEM
C           *** FIRST WORK WITH RHS
               YMW(I)  = 1.0 + DR*XX1M(I)
               C(I,4)  = U(I)*YMW(I)  +  (U(I-1)+U(I+1))*YLRWS
C           *** WORK WITH LHS
               C(I,1)  = XLRWS
               C(I,2)  = 2.0 - YMW(I)
               C(I,3)  = XLRWS
   10    CONTINUE
C     *** COMPUTE LHS & RHS ELEMENTS IN SEDIMENT
         IFACEP = IFACE + 1
         NM1 = N - 1
         DO 20 I=IFACEP,NM1
C           *** RHS
               C(I,4)  = U(I)*YMS + (U(I-1)+U(I+1))*YLRWS
C           *** LHS
               C(I,1)  = XLRWS
               C(I,2)  = XMS
               C(I,3)  = XLRWS
   20    CONTINUE
C
C     *** IF ENTIRE SEGMENT LEVEL GO TO 50
```

```
          IF ( ISLOPE.EQ.2 ) GO TO 50
C
C     *** INTERFACE SLOPES EITHER UP OR DOWN
C     *** CALCULATE CONSTANTS FOR COMPUTING MATRIX ELEMENTS
          THETA = ABS (THETA)
          SINE = SIN (THETA)
          COSE = COS (THETA)
          DELTA = XX10  + XX11*(XX8+XX9)*SINE*COSE
          DELIN = 1.0 / DELTA
          BEDA1 = DELIN * XX12 * RHO1
          GAMMA1 = DELIN *  (XX12*(RHO2*COSE*COSE+RHO1*SINE*SINE) +
     *             XX11*SINE*COSE/DZ)
          BEDA2 = DELIN   *  (XX12*(RHO1*COSE*COSE+RHO2*SINE*SINE) +
     *             XX11*SINE*COSE/DZ)
          GAMMA2 = DELIN * XX12 * RHO2
          ZZ1 = DELIN *  (RHO1*SINE*SINE + RHO2*COSE*COSE +
     *           XX8*SINE*COSE*XX11)
          ZZ2 = DELIN *  (RHO1*A2 -  (COSE-XX8*SINE)*XX9*XX11*EYE*XKO*
     *           SINE)
          ZZ3 = DELIN * RHO2
          ZZ4 = DELIN *  (  A2 *  ( RHO1*COSE*COSE + RHO2*SINE*SINE
     *           + XX8*SINE*COSE*XX11 )  -
     *           ( COSE  + XX8*SINE  )*XX9*XX11*EYE*XKO*SINE  )
          ZZ5 = 0.5*DR*ZZ1
          ZZ6 = 1.0 + 0.5*DR*(ZZ2-BEDA1-GAMMA1)
          ZZ7 = -0.5*DR*ZZ3
          ZZ8 = 1.0 - 0.5*DR*(ZZ4-BEDA2-GAMMA2)
          ZZ9 = 2.0 - ZZ8
          ZZ10 = 2.0 -  ZZ6
C
C         *** IF BOTTOM SLOPES UP, GO TO 40
              IF ( ISLOPE.EQ.3 .OR. ISLOPE.EQ.5 ) GO TO 40
C         *** BOTTOM SLOPES DOWN
              IFACE2 = IFACEP
C             *** COMPUTE OFF-DIAGONAL, Y MATRIX ELEMENTS ON INTERFACE
                  YLI = 0.5 * DR * GAMMA1
                  YRI = 0.5 * DR * BEDA1
C             *** COMPUTE MAIN DIAGONAL, Y MATRIX ELEMENT ON INTERFACE
                  YMI = A(IFACE) * ZZ5  +  ZZ6
C             *** COMPUTE INTERFACE ELEMENT IN RHS COLUMN VECTOR
                  C(IFACE,4)  = U(IFACEM)*YLI + U(IFACE)*YMI +
     *                          U(IFACEP)*YRI
C             *** COMPUTE X MATRIX ELEMENTS ON INTERFACE
                  XLI = -0.5 * DR * GAMMA2
                  XMI = A(IFACE2) * ZZ7  +  ZZ8
                  XRI = -0.5 * DR * BEDA2
C                 *** IF MODIFIED BOTTOM THEN NO NEED TO ADJUST LHS
                      IF ( ISLOPE.EQ.4 ) GO TO 45
```

```
                C(IFACE2,1) = XLI
                C(IFACE2,2) = XMI
                C(IFACE2,3) = XRI
C           *** COMPUTE X MATRIX ELEMENTS ONE ROW ABOVE INTERFACE
                C(IFACE,1) = XLRWS
                C(IFACE,2) = 1.0 - DR * XX1M(IFACE)
                C(IFACE,3) = XLRWS
            GO TO 60
C
C       *** BOTTOM SLOPES UP
 40         IFACE2 = IFACEM
C           *** COMPUTE OFF-DIAGONAL, Y MATRIX ELEMENTS ON INTERFACE
                YLI = 0.5 * DR * GAMMA2
                YRI = 0.5 * DR * BEDA2
C           *** COMPUTE MAIN DIAGONAL, Y MATRIX ELEMENT ON INTERFACE
                YMI = - A(IFACE) * ZZ7  +  ZZ9
C           *** COMPUTE INTERFACE ELEMENT IN RHS COLUMN VECTOR
                C(IFACE,4) = U(IFACEM)*YLI + U(IFACE)*YMI +
     *                       U(IFACEP)*YRI
C           *** COMPUTE X MATRIX ELEMENTS ON INTERFACE
                XLI = -0.5 * DR * GAMMA1
                XMI = - A(IFACE2) * ZZ5  +  ZZ10
                XRI = -0.5 * DR * BEDA1
C               *** IF MODIFIED BOTTOM THEN NO NEED TO ADJUST LHS
                    IF ( ISLOPE.EQ.5 ) GO TO 45
                C(IFACE2,1) = XLI
                C(IFACE2,2) = XMI
                C(IFACE2,3) = XRI
C           *** COMPUTE X MATRIX ELEMENTS ONE ROW BELOW INTERFACE
                C(IFACE,1) = XLRWS
                C(IFACE,2) = XMS
                C(IFACE,3) = XLRWS
            GO TO 60
C
C       *** SAVE INTERFACE VALUES ON SLOPING SECTION
 45         YLIZ = YLI
            YRIZ = YRI
            XLIZ = XLI
            XRIZ = XRI
C
C       *** SEGMENT LEVEL, COMPUTE MATRIX ELEMENTS CN INTERFACE
 50         IFACE2 = IFACE
            YLI = DR * XX6
            YMI = 1.0 + DR * ( XX1M(IFACE)/XX4 + XX5 )
            YRI = DR * XX7
            C(IFACE,4) = U(IFACEM)*YLI + U(IFACE)*YMI + U(IFACEP)*YRI
            C(IFACE,1) = -YLI
            C(IFACE,2) = 2.0 - YMI
```

```
            C(IFACE,3) = -YRI
C           *** SAVE INTERFACE VALUES ON LEVEL SECTION
               YLIV = YII
               YRIV = YRI
C
 60   CONTINUE
C
C     *** COMPUTE ARTIFICIAL ATTENUATION MATRIX
C     *** SEE AESD PE MODEL BY BROCK  -  NORDA TECH NOTE 12  -  JAN 78
C        *** CALCULATE GRID POINT AT TOP OF ART ATTENUATION LAYER
            IA = INT ( ZABLYR/DZ + 0.01 )
C        *** CALCULATE NUMBER OF GRID POINTS IN ART ATTENUATION LAYER
            NA = N - IA
C        *** CALCULATE ATTENUATION MATRIX
            DO 70 I=1,NA
               TEMP = 3.0 * (I-NA)  /  NA
               ATT(I)  = EXP(-0.01*DR*EXP(-(TEMP*TEMP)))
 70         CONTINUE
C
C     *** SOLVE FOR SOLUTION FIELD AT RANGE RA2
         CALL TRIDG (C,U,N,CR,CTWO)
C
C     *** APPLY ARTIFICAL ATTENUTION
         CALL ATTENU (U,ATT,IA,NA)
C
      RA2P = RA2 + 0.5
C     *** TIME TO WRITE?
         IF ( RA2P.GE. XWR )  CALL WRITE2
C     *** TIME TO PRINT?
         IF ( RA2P.GE. XPR )  CALL PRTWT2
C
C     *** UPDATE INTERFACE POINTER
         IFACE = IFACE2
C
      RETURN
      END
C
C
C
C
      SUBROUTINE TRIDG (C,U,N,CR,CTWO)
C
C     *** THIS SUBROUTINE SOLVES A SET OF  N - 1  (NM1)  LINEAR
C     *** SIMULTANEOUS EQUATIONS HAVING A TRIDIAGONAL COEFFICIENT
C     *** MATRIX.  MATRIX ELEMENTS IN THE LOWER DIAGONAL, MAIN DIAGONAL
C     *** AND UPPER DIAGONAL ARE STORED IN C(I,1), C(I,2) AND C(I,3)
C     *** RESPECTIVELY.  THE RHS COLUMN VECTOR IS STORED IN C(I,4).
C     *** THE SOLUTION FIELD IS STORED IN U(I).
```

```
C       ***      (1)  THE INDEX I REFERS TO ROW NUMBER.
C       ***      (2)  WE NEED ONLY SOLVE AN  NM1 X NM1  SYSTEM (RATHER THAN
C       ***           AN  N X N  SYSTEM) BECAUSE U(N) IS KNOWN:  U(N)=0.0
C       ***      (3)  THE SUBROUTINE IS A MODIFIED VERSION OF SUBROUTINE
C       ***           TRIDG FROM:
C       ***              "APPLIED NUMERICAL ANALYSIS"   ( SECOND EDITION )
C       ***               BY:  CURTIS F. GERALD
C       ***               PUBLISHED BY ADDISON-WESLEY PUBLISING CO., 1980
C       ***      (4)  THE MAIN MODIFICATIONS TO THE ROUTINE IN THE TEXT
C       ***           INVOLVED:
C       ***         (A)   INTRODUCING ARRAYS CTWO AND CR TO PRESERVE THE
C       ***               ORIGINAL VALUES IN C(I,2) AND TO MAKE THE ROUTINE
C       ***               MORE EFFICIENT.  THIS RESULTS IN A CONSIDERABLE
C       ***               SAVINGS IN EXECUTION TIME FOR THE CASE OF A
C       ***               HORIZONTAL BOTTOM.   (SEE SUBROUTINE TRIDL)
C       ***         (B)   MODIFYING THE ROUTINE TO SOLVE AN  NM1 X NM1
C       ***               SYSTEM.
C       ***       (5)  SEE PAGES 129 AND 133 IN THE TEXT FOR FURTHER INFO.
C
      COMPLEX C(5000,4), U(5000), CR(5000), CTWO(5000)
C
      NM1 = N - 1
      NM2 = N - 2
      CTWO(1) = C(1,2)
      DO 10 I=2,NM1
         CR(I)   = C(I,1)  / CTWO(I-1)
         CTWO(I) = C(I,2)  - CR(I)  * C(I-1,3)
         C(I,4)  = C(I,4)  - CR(I)  * C(I-1,4)
 10   CONTINUE
C
      U(N)  = 0.0
C
C     *** NOW PERFORM BACK SUBSTITUTION
C
      U(NM1) = C(NM1,4)  / CTWO(NM1)
      DO 20  I=1,NM2
         M = NM1 - I
         U(M) =  ( C(M,4)-C(M,3)*U(M+1)  ) / CTWO(M)
 20   CONTINUE
C
      RETURN
      END
C
C
C
C
      SUBROUTINE TRIDL  (C,U,N,CR,CTWO)
C
```

```
C       *** THIS SUBROUTINE IS A MODIFIED VERSION OF SUBROUTINE TRIDG
C       *** FROM THE IFD PROGRAM. SUBROUTINE TRIDG IS IN TURN A MODIFIED
C       *** VERSION OF TRIDG AS PER THE REFERENCE BELOW.
C       *** THE SUBROUTINE SOLVES A SET OF  N - 1  (NM1)  LINEAR
C       *** SIMULTANEOUS EQUATIONS HAVING A TRIDIAGONAL COEFFICIENT
C       *** MATRIX.  MATRIX ELEMENTS IN THE LOWER DIAGONAL, MAIN DIAGONAL
C       *** AND UPPER DIAGONAL ARE STORED IN C(I,1), C(I,2) AND C(I,3)
C       *** RESPECTIVELY.  THE RHS COLUMN VECTOR IS STORED IN C(I,4).
C       *** THE SOLUTION FIELD IS STORED IN U(I).
C       ***     (1)  THE INDEX I REFERS TO ROW NUMBER.
C       ***     (2)  WE NEED ONLY SOLVE AN  NM1 X NM1  SYSTEM (RATHER THAN
C       ***          AN  N X N  SYSTEM) BECAUSE U(N) IS KNOWN:  U(N)=0.0
C       ***     (3)  THE SUBROUTINE IS A MODIFIED VERSION OF IFD SUB-
C       ***          ROUTINE TRIDG WHICH IN TURN IS A MODIFIED VERSION
C       ***          OF SUBROUTINE TRIDG AS PER:
C       ***            "APPLIED NUMERICAL ANALYSIS"   ( SECOND EDITION )
C       ***             BY:  CURTIS F. GERALD
C       ***             PUBLISHED BY ADDISON-WESLEY PUBLISING CO., 1980
C       ***     (4)  THE ONLY MODIFICATION TO IFD SUBROUTINE TRIDG IS
C       ***          THAT TRIDL DOES NOT RECALCULATE CTWO AND CR, BUT
C       ***          BUT RATHER TRIDL USES THE ARRAY VALUES CALCULATED
C       ***          BY TRIDG.  THIS RESULTS IN A CONSIDERABLE SAVINGS
C       ***          IN EXECUTION TIME FOR THE CASE OF A HORIZONTAL
C       ***          BOTTOM.
C
        COMPLEX C(5000,4), U(5000), CR(5000), CTWO(5000)
C
        NM1 = N - 1
        NM2 = N - 2
        DO 10 I=2,NM1
           C(I,4) = C(I,4) - CR(I) * C(I-1,4)
  10    CONTINUE
C
        U(N)  = 0.0
C
C       *** NOW PERFORM BACK SUBSTITUTION
C
        U(NM1) = C(NM1,4) / CTWO(NM1)
        DO 20 I=1,NM2
           M = NM1 - I
           U(M) = ( C(M,4)-C(M,3)*U(M+1) ) / CTWO(M)
  20    CONTINUE
C
        RETURN
        END
C
C
C
```

```
C
      SUBROUTINE DOWN
C
C     THIS SUBROUTINE UPDATES THE RHS & LHS OF THE EQUATION AND
C     SOLVES FOR THE SOLUTION FIELD AT RA2.
C        (1)  THE RHS COLUMN VECTOR VALUES ARE STORED IN C(I,4)
C        (2)  THE INTERFACE AT RANGE RA1 IS AT GRIDPOINT IFACE
C        (3)  THE INTERFACE AT RANGE RA2 IS AT GRIDPOINT IFACE2
C                ( WHERE IFACE2 = IFACE + 1 )
C
      COMPLEX A,A2,C,CR,CTWO,EYE,
     *        XLI,XLIZ,XLRWS,XMI,XMS,XRI,XRIZ,
     *        XX1,XX2,XX3,XX5,XX6,XX7,XX8,XX9,XX12,XX1M,
     *        YLI,YLIV,YLIZ,YLRWS,YMI,YMS,YMW,YRI,YRIV,YRIZ,
     *        U,ZZ5,ZZ6,ZZ7,ZZ8,ZZ9,ZZ10
      COMMON /IN/ IA,IBOT1,IFACE,IPZ,ISLOPE,ISTEP,IWZ,N,NA,NBOT,NM1,
     *        NSTEP,NSTEP1,NSVP,NWMAX,NXLFS
      COMMON /REAL/ ALPHA,ATT(5000),BETA1,BETA2,ER(101),BZ(101),CO,
     *        CSVE(101),C2,CWATER(5000),DR,DRLVL,DRMAX,DZ,FRQ,PDR,PDZ,
     *        R1,RA1,RA2,RHO1,RHO2,RMAX,THETA,XK0,XLAMDA,XPR,XX4,XX10,
     *        XX11,XWR,WDR,ZLYR1,ZLYR2,ZR,ZS,ZSVE(101),ZABLYR,
     *        PRTIN,PRTOT
      COMMON /CPIX/  A(5000),A2,C(5000,4),CR(5000),CTWO(5000),
     *        EYE,XLI,XLIZ,XLRWS,XMI,XMS,XRI,XRIZ,
     *        XX1,XX2,XX3,XX5,XX6,XX7,XX8,XX9,XX12,XX1M(5000),
     *        YLI,YLIV,YLIZ,YLRWS,YMI,YMS,YMW(5000),YRI,YRIV,YRIZ,
     *        U(5000),ZZ5,ZZ6,ZZ7,ZZ8,ZZ9,ZZ10
C
C     *** UPDATE IFACE2
          IFACE2 = IFACE + 1
C     *** UPDATE Y MATRIX, MAIN DIAGONAL, INTERFACE ELEMENT
          YMI = A(IFACE) * ZZ5  +  ZZ6
C     *** UPDATE Y MATRIX, MAIN DIAGONAL, WATER ELEMENT, ONE ROW
C     *** ABOVE INTERFACE
          YMW(IFACE-1)  =  1.0 + DR * XX1M(IFACE-1)
C
C     *** UPDATE RHS
          CALL RHS
C     *** UPDATE LHS
C        *** UPDATE X MATRIX ELEMENTS ONE ROW ABOVE INTERFACE
             C(IFACE,1)  =  XLRWS
             C(IFACE,2)  =  1.0 - DR*XX1M(IFACE)
             C(IFACE,3)  =  XLRWS
C        *** UPDATE X MATRIX ELEMENTS ON INTERFACE
             C(IFACE2,1)  = XLI
             C(IFACE2,2)  = A(IFACE2) * ZZ7  +  ZZ8
             C(IFACE2,3)  = XRI
C
```

```
C     *** SOLVE THE TRIDIAGONAL SYSTEM
          CALL TRIDG (C,U,N,CR,CTWO)
C     *** UPDATE IFACE
          IFACE = IFACE2
      RETURN
      END
C
C
C
C
      SUBROUTINE UP
C
C     THIS SUBROUTINE UPDATES THE RHS & LHS OF THE EQUATION AND
C     SOLVES FOR THE SOLUTION FIELD AT RA2.
C        (1)  THE RHS COLUMN VECTOR VALUES ARE STORED IN C(I,4)
C        (2)  THE INTERFACE AT RANGE RA1 IS AT GRIDPOINT IFACE
C        (3)  THE INTERFACE AT RANGE RA2 IS AT GRIDPOINT IFACE2
C                ( WHERE IFACE2 = IFACE - 1 )
C
      COMPLEX A,A2,C,CR,CTWO,EYE,
     *          XLI,XLIZ,XLRWS,XMI,XMS,XRI,XRIZ,
     *          XX1,XX2,XX3,XX5,XX6,XX7,XX8,XX9,XX12,XX1M,
     *          YLI,YLIV,YLIZ,YLRWS,YMI,YMS,YMW,YRI,YRIV,YRIZ,
     *          U,ZZ5,ZZ6,ZZ7,ZZ8,ZZ9,ZZ10
      COMMON /IN/ IA,IBOT1,IFACE,IPZ,ISLOPE,ISTEP,IWZ,N,NA,NBOT,NM1,
     *          NSTEP,NSTEP1,NSVP,NWMAX,NXLFS
      COMMON /REAL/ ALPHA,ATT(5000),BETA1,BETA2,BR(101),BZ(101),CO,
     *          CSVP(101),C2,CWATER(5000),DR,DRLVL,DRMAX,DZ,FRQ,PDR,PDZ,
     *          R1,RA1,RA2,RHO1,RHO2,RMAX,THETA,XK0,XLAMDA,XPR,XX4,XX10,
     *          XX11,XWR,WDR,ZLYR1,ZLYR2,ZR,ZS,ZSVP(101),ZABLYR,
     *          PRTIN,PRTOT
      COMMON /CPLX/  A(5000),A2,C(5000,4),CR(5000),CTWO(5000),
     *          EYE,XLI,XLIZ,XLRWS,XMI,XMS,XRI,XRIZ,
     *          XX1,XX2,XX3,XX5,XX6,XX7,XX8,XX9,XX12,XX1M(5000),
     *          YLI,YLIV,YLIZ,YLRWS,YMI,YMS,YMW(5000),YRI,YRIV,YRIZ,
     *          U(5000),ZZ5,ZZ6,ZZ7,ZZ8,ZZ9,ZZ10
C
C     *** UPDATE IFACE2
          IFACE2 = IFACE - 1
C     *** UPDATE Y MATRIX, MAIN DIAGONAL, INTERFACE ELEMENT
          YMI = -A(IFACE) * ZZ7  +  ZZ9
C     *** UPDATE RHS
          CALL RHS
C     *** UPDATE LHS
C        *** UPDATE X MATRIX ELEMENTS ONE ROW BELOW INTERFACE
             C(IFACE,1)  = XLRWS
             C(IFACE,2)  = XMS
             C(IFACE,3)  = XLRWS
```

```
C         *** UPDATE X MATRIX ELEMENTS ON INTERFACE
                 C(IFACE2,1) = XLI
                 C(IFACE2,2) = -A(IFACE2) * ZZ5  +  ZZ10
                 C(IFACE2,3) = XRI
C
C       *** SOLVE THE TRDIAGCNAL SYSTEM
            CALL TRIDG (C,U,N,CR,CTWO)
C       *** UPDATE IFACE
            IFACE = IFACE2
        RETURN
        END
C
C
C
C
        SUBROUTINE SSLOPE
C
C       *** THIS SUBROUTINE IS CALLED TO ADVANCE THE SOLUTICN FIELD
C       *** FOR THE CASE OF A MODIFIED BOTTOM.
C       ***    (1)   THIS CASE OCCURS WHEN THE BOTTOM SLOPE IS TOO SMALL
C       ***          FOR THE MAXIMUM RANGE STEP.
C       ***    (2)   THIS SUBROUTINE WORKS FOR BOTH A DCWNSLOPE AND
C       ***          UPSLOPE MCDIFIED BOTTOM.
C       ***    (3)   THE SUBROUTINE DETERMINES WHICH OF THE FCLLOWING
C       ***          THREE TYPES OF BOTTOM SECTIONS NEEDS TO BE CONSIDERED:
C       ***            (A)  LEVEL SECTION FOLLOWS LEVEL SECTION
C       ***            (B)  LEVEL SECTION FOLLOWS SLOPING SECTION
C       ***            (C)  SLOPING SECTION.
C       ***    (4)   AFTER DETERMINING WHICH OF THE THREE TYPES OF BOTTCM
C       ***          SECTIONS IS APPROPRIATE, THE SUBROUTINE MAKES MATRIX
C       ***          ELEMENT CHANGES AS REQUIRED AND CALLS ON OTHER
C       ***          SUBROUTINES TO ADVANCE THE SOLUTICN.
C
        CCMPLEX A,A2,C,CR,CTWO,EYE,
       *        XLI,XLIZ,XLRWS,XMI,XMS,XRI,XRIZ,
       *        XX1,XX2,XX3,XX5,XX6,XX7,XX8,XX9,XX12,XX1M,
       *        YLI,YLIV,YLIZ,YLRWS,YMI,YMS,YMW,YRI,YRIV,YRIZ,
       *        U,ZZ5,ZZ6,ZZ7,ZZ8,ZZ9,ZZ10
        COMMON /IN/ IA,IBOT1,IFACE,IPZ,ISLOPE,ISTEP,IWZ,N,NA,NBOT,NM1,
       *        NSTEP,NSTEP1,NSVP,NWMAX,NXLFS
        CCMMON /REAL/ ALPHA,ATT(5000),BETA1,BETA2,BR(101),BZ(101),CO,
       *        CSVP(101),C2,CWATER(5000),DR,DRLVL,DRMAX,DZ,FRQ,PDR,PDZ,
       *        R1,RA1,RA2,RHO1,RHO2,RMAX,THETA,XK0,XLAMDA,XPR,XX4,XX10,
       *        XX11,XWR,WDR,ZLYR1,ZLYR2,ZR,ZS,ZSVP(101),ZABLYR,
       *        PRTIN,PRTCT
        CCMMON /CPLX/  A(5000),A2,C(5000,4),CR(5000),CTWO(5000),
       *        EYE,XLI,XLIZ,XLRWS,XMI,XMS,XRI,XRIZ,
       *        XX1,XX2,XX3,XX5,XX6,XX7,XX8,XX9,XX12,XX1M(5000),
```

```
     *     YLI,YLIV,YLIZ,YLRWS,YMI,YMS,YMW(5000),YRI,YRIV,YRIZ,
     *     U(5000),ZZ5,ZZ6,ZZ7,ZZ8,ZZ9,ZZ10
C
C    *** IS THIS A LEVEL SECTION FOLLOWING A SLOPING SECTION?
        IF (ISTEP.EQ.NXLFS)   GO TO 20
C.   *** NO
C
C    *** IS THIS A SLOPING SECTION?
        ITEST = NXLFS - ISTEP
        IF (ITEST.EQ.1) GO TO 30
C    *** NO
C
C    *** LEVEL SECTION FOLLOWS A LEVEL SECTION
        CALL LEVEL
        GO TO 50
C
C    *** LEVEL SECTION FOLLOWS A SLOPING SECTION
C       *** UPDATE NXLFS
 20         NXLFS = NXLFS + NSTEP1
C       *** IF LAST SECTION SLOPED DOWN, UPDATE Y MATRIX ELEMENT,
C       *** MAIN DIAGONAL, IN WATER, ONE ROW ABOVE INTERFACE
              IF ( ISLOPE.EQ.5 ) GO TO 25
              YMW(IFACE-1) = 1.0 + DR * XX1M(IFACE-1)
C       *** UPDATE RHS INTERFACE ELEMENTS
 25         YLI = YLIV
            YMI = 1.0 + DR * ( XX1M(IFACE)/XX4 + XX5 )
            YRI = YRIV
C       *** UPDATE LHS
            C(IFACE,1) = -YLI
            C(IFACE,2) = 2.0 - YMI
            C(IFACE,3) = -YRI
C       *** SOLVE SYSTEM
            CALL RHS
            CALL TRIDG(C,U,N,CR,CTWO)
        GO TO 50
C
C    *** SLOPING SECTION
C       *** UPDATE INTERFACE ELEMENTS
 30         YLI = YLIZ
            YRI = YRIZ
            XLI = XLIZ
            XRI = XRIZ
C       *** SOLVE SYSTEM AS APPROPRIATE
            IF (ISLOPE.EQ.4) CALL DOWN
            IF (ISLOPE.EQ.5) CALL UP
C
 50  CONTINUE
     RETURN
```

```
      END
C
C
C
C
      SUBROUTINE RHS
C
C     THIS SUBROUTINE MULTIPLIES TRIDIAGONAL MATRIX Y TIMES SCLUTICN
C     FIELD U TO CBTAIN AN UPDATED RHS.
C        (1)  THE RHS COLUMN VECTOR VALUES ARE STORED IN C(I,4).
C
      CCMPLEX A,A2,C,CR,CTWO,EYE,
     *        XLI,XLIZ,XLRWS,XMI,XMS,XRI,XRIZ,
     *        XX1,XX2,XX3,XX5,XX6,XX7,XX8,XX9,XX12,XX1M,
     *        YLI,YLIV,YLIZ,YLRWS,YMI,YMS,YMW,YRI,YRIV,YRIZ,
     *        U,ZZ5,ZZ6,ZZ7,ZZ8,ZZ9,ZZ10
      CCMMON /IN/ IA,IBOT1,IFACE,IPZ,ISLOPE,ISTEP,IWZ,N,NA,NBOT,NM1,
     *        NSTEP,NSTEP1,NSVP,NWMAX,NXLFS
      CCMMON /REAL/ ALPHA,ATT(5000),BETA1,BETA2,BR(101),BZ(101),C0,
     *        CSVP(101),C2,CWATER(5000),DR,DRLVL,DRMAX,DZ,FRQ,PDR,PDZ,
     *        R1,RA1,RA2,RHO1,RHO2,RMAX,THETA,XK0,XLAMDA,XPR,XX4,XX10,
     *        XX11,XWR,WDR,ZLYR1,ZLYR2,ZR,ZS,ZSVP(101),ZABLYR,
     *        PRTIN,PRTOT
      CCMMON /CPIX/  A(5000),A2,C(5000,4),CR(5000),CTWO(5000),
     *        EYE,XLI,XLIZ,XLRWS,XMI,XMS,XRI,XRIZ,
     *        XX1,XX2,XX3,XX5,XX6,XX7,XX8,XX9,XX12,XX1M(5000),
     *        YLI,YLIV,YLIZ,YLRWS,YMI,YMS,YMW(5000),YRI,YRIV,YRIZ,
     *        U(5000),ZZ5,ZZ6,ZZ7,ZZ8,ZZ9,ZZ10
C
C     *** UPDATE IFACEM & IFACEP
          IFACEM = IFACE - 1
          IFACEP = IFACE + 1
C     *** UPDATE RHS
          C(1,4) = U(1)  * YMW(1)   +  U(2) * YLRWS
          DO 10 I=2,IFACEM
             C(I,4) = U(I) * YMW(I)   +   (U(I-1)+U(I+1)) * YLRWS
 10       CONTINUE
          C(IFACE,4) = U(IFACEM)*YLI + U(IFACE)*YMI + U(IFACEP)*YRI
          DO 20 I=IFACEP,NM1
             C(I,4) = U(I)*YMS + (U(I-1)+U(I+1))*YLRWS
 20       CONTINUE
C
      RETURN
      END
C
C
C
C
```

```
      SUBROUTINE LEVEL
C
C     THIS SUBROUTINE UPDATES THE RHS OF THE EQUATION AND SOLVES
C     FOR THE SOLUTION FIELD AT RANGE RA2.
C        (1)  THE RHS COLUMN VECTOR VALUES ARE STORED IN C(I,4).
C        (2)  FOR THE LEVEL INTERFACE THE LHS TRIDIAGINAL MATRIX
C             ELEMENTS NEED NOT BE UPDATED.
C
      COMPLEX A,A2,C,CR,CTWO,EYE,
     *        XLI,XLIZ,XLRWS,XMI,XMS,XRI,XRIZ,
     *        XX1,XX2,XX3,XX5,XX6,XX7,XX8,XX9,XX12,XX1M,
     *        YLI,YLIV,YLIZ,YLRWS,YMI,YMS,YMW,YRI,YRIV,YRIZ,
     *        U,ZZ5,ZZ6,ZZ7,ZZ8,ZZ9,ZZ10
      COMMON /IN/ IA,IBOT1,IFACE,IPZ,ISLOPE,ISTEP,IWZ,N,NA,NBOT,NM1,
     *        NSTEP,NSTEP1,NSVP,NWMAX,NXLFS
      COMMON /REAL/ ALPHA,ATT(5000),BETA1,BETA2,BR(101),BZ(101),C0,
     *        CSVP(101),C2,CWATER(5000),DR,DRLVL,DRMAX,DZ,FRQ,PDR,PDZ,
     *        R1,RA1,RA2,RHO1,RHO2,RMAX,THETA,XK0,XLAMDA,XPR,XX4,XX10,
     *        XX11,XWR,WDR,ZLYR1,ZLYR2,ZR,ZS,ZSVP(101),ZABLYR,
     *        PRTIN,PRTOT
      COMMON /CPIX/  A(5000),A2,C(5000,4),CR(5000),CTWO(5000),
     *        EYE,XLI,XLIZ,XLRWS,XMI,XMS,XRI,XRIZ,
     *        XX1,XX2,XX3,XX5,XX6,XX7,XX8,XX9,XX12,XX1M(5000),
     *        YLI,YLIV,YLIZ,YLRWS,YMI,YMS,YMW(5000),YRI,YRIV,YRIZ,
     *        U(5000),ZZ5,ZZ6,ZZ7,ZZ8,ZZ9,ZZ10
C
C      *** UPDATE RHS
          CALL RHS
C
C      *** SOLVE THE TRIDIAGONAL SYSTEM
          CALL TRIDL (C,U,N,CR,CTWO)
C
      RETURN
      END
C
C
C
C
      SUBROUTINE PRTWT2
C
C     THIS SUBROUTINE IS USED TO OUTPUT DATA IN A FORMAT WHICH IS
C     REDUCED IN VOLUME AND COMPATIBLE FOR THE WAVENUMBER TECHNIQUE
C     (WT)
C
C
C        (1)  THIS SUBROUTINE IS EFFECTIVELY THE CONTINUATION OF
C             SUBROUTINE PRTWT1.
C        (2)  THE FILE CREATED CORRESPONDS TO UNIT FILE NUMBER:
```

```
C            NPOUT = 55
C        (3) THE FILENAME AND FILETYPE FOR THIS FILE ARE:
C                IFDOUT     PRINTER
C            NPOUT1 = 60
C        (4) THE FILENAME AND FILETYPE FOR THIS FILE ARE:
C                IFDOUTA     PRINTER
C
      COMPLEX A,A2,C,CR,CTWO,EYE,
     *        XLI,XLIZ,XLRWS,XMI,XMS,XRI,XRIZ,
     *        XX1,XX2,XX3,XX5,XX6,XX7,XX8,XX9,XX12,XX1M,
     *        YLI,YLIV,YLIZ,YLRWS,YMI,YMS,YMW,YRI,YRIV,YRIZ,
     *        U,ZZ5,ZZ6,ZZ7,ZZ8,ZZ9,ZZ10
      COMMON /IN/ IA,IBOT1,IFACE,IPZ,ISLOPE,ISTEP,IWZ,N,NA,NBOT,NM1,
     *        NSTEP,NSTEP1,NSVP,NWMAX,NXLFS
      COMMON /REAL/ ALPHA,ATT(5000),BETA1,BETA2,ER(101),BZ(101),C0,
     *        CSVP(101),C2,CWATER(5000),DR,DRLVL,DRMAX,DZ,FRQ,PDR,PDZ,
     *        R1,RA1,RA2,RHO1,RHO2,RMAX,THETA,XK0,XLAMDA,XPR,XX4,XX10,
     *        XX11,XWR,WDR,ZLYR1,ZLYR2,ZR,ZS,ZSVP(101),ZABLYR,
     *        PRTIN,PRTOT
      COMMON /CPIX/  A(5000),A2,C(5000,4),CR(5000),CTWO(5000),
     *        EYE,XLI,XLIZ,XLRWS,XMI,XMS,XRI,XRIZ,
     *        XX1,XX2,XX3,XX5,XX6,XX7,XX8,XX9,XX12,XX1M(5000),
     *        YLI,YLIV,YLIZ,YLRWS,YMI,YMS,YMW(5000),YRI,YRIV,YRIZ,
     *        U(5000),ZZ5,ZZ6,ZZ7,ZZ8,ZZ9,ZZ10
      DATA NPOUT/55/,NPOUT1/60/
          ZWHOLD = 0.0
          ZLHOLD = 0.0
C
C     *** INDICATE TO USER HOW FAR SOLUTION FIELD HAS PROGRESSED
C             BY INDICATING 1 KM STEPS
          LPRTCK = MOD(INT(RA2),1000)
          IF(LPRTCK .EQ. 0) WRITE(6,10) RA2
   10 FORMAT(/,' THE PROGRAM HAS REACHED RANGE R =',F10.3,' METERS.')
C
C     *** PRINT RANGE, U (REAL) AND U (IMAGINARY)
C
          DO 20 I=IPZ,N,IPZ
          ZI = FLOAT(I)*DZ
          ZIH = ZI+(FLOAT(I+IPZ)*DZ)
          IF(ZI.EQ.ZR)LCHK = I
          IF((ZI.LT.ZR) .AND. (ZIH.GT.ZR))
     *    LCHK = I+1
          IF(ZI.EQ.ZLYR1)ZWHOLD = CABS(U(I))
          IF((ZI.LT.ZLYR1) .AND. (ZIH.GT.ZLYR1))
     *    ZWHOLD = CABS(U(I+1))
          IF(ZI.EQ.ZABLYR)ZLHOLD = CABS(U(I))
          IF((ZI.LT.ZABLYR) .AND. (ZIH.GT.ZABLYR))
     *    ZLHOLD = CABS(U(I+1))
```

```
 20         CONTINUE
 800    FORMAT(T5,F9.2,2(5X,E12.5))
C
C       LIMIT THE PRINT OUT
C
        IF((RA2 .LT. PRTIN) .OR. (RA2 .GT. PRTOT)) GO TO 1000
        WRITE(NPOUT,800) RA2,U(LCHK)
        IF((ZWHOLD .LE. 0.).OR.(ZLHOLD .LE. 0.)) GO TO 1000
        ZTLW  = -20.0*ALOG10(ZWHOLD)+10.0*ALOG10(RA2)
        ZTLL  = -20.0*ALOG10(ZLHOLD)+10.0*ALOG10(RA2)
        WRITE(NPOUT1,900) RA2,ZLYR1,ZABLYR,ZTLW,ZTLL
 900    FORMAT(T5,3(5X,F9.2),2(2X,F9.4))
 1000   CONTINUE
        XPR = XPR + DPR
C
C
        RETURN
        END
C
C
C
C
        SUBROUTINE WRITE2
C
C          (1)   THIS SUBROUTINE IS EFFECTIVELY THE CONTINUATION OF
C                SUBROUTINE WRITE1
C          (2)   THE SUBROUTINE WRITES RANGE, RECEIVER DEPTH AND U(I)
C                WHEN CALLED.  IT THEN UPDATES THE NEXT WRITE RANGE (XWR).
C          (3)   THE FILE WRITTEN INTO CORRESPONDS TO UNIT FILE NUMBER:
C                NOU = 52.
C          (4)   THE FILENAME AND FILETYPE FOR THIS FILE ARE:
C                     IFDOUT   PLOTTER
C
        COMPLEX A,A2,C,CR,CTWO,EYE,
     *          XLI,XLIZ,XLRWS,XMI,XMS,XRI,XRIZ,
     *          XX1,XX2,XX3,XX5,XX6,XX7,XX8,XX9,XX12,XX1M,
     *          YLI,YLIV,YLIZ,YLRWS,YMI,YMS,YMW,YRI,YRIV,YRIZ,
     *          U,ZZ5,ZZ6,ZZ7,ZZ8,ZZ9,ZZ10
        COMMON /IN/ IA,IBOT1,IFACE,IPZ,ISLOPE,ISTEP,IWZ,N,NA,NBOT,NM1,
     *          NSTEP,NSTEP1,NSVP,NWMAX,NXLFS
        COMMON /REAL/ ALPHA,ATT(5000),BETA1,BETA2,BR(101),BZ(101),CO,
     *          CSVP(101),C2,CWATER(5000),DR,DRLVL,DRMAX,DZ,FRQ,PDR,PDZ,
     *          R1,RA1,RA2,RHO1,RHO2,RMAX,THETA,XK0,XLAMDA,XPR,XX4,XX10,
     *          XX11,XWR,WDR,ZLYR1,ZLYR2,ZR,ZS,ZSVP(101),ZABLYR,
     *          PRTIN,PRTOT
        COMMON /CPIX/  A(5000),A2,C(5000,4),CR(5000),CTWO(5000),
     *          EYE,XLI,XLIZ,XLRWS,XMI,XMS,XRI,XRIZ,
     *          XX1,XX2,XX3,XX5,XX6,XX7,XX8,XX9,XX12,XX1M(5000),
```

```
     *         YLI,YLIV,YLIZ,YLRWS,YMI,YMS,YMW(5000),YRI,YRIV,YRIZ,
     *         U(5000),ZZ5,ZZ6,ZZ7,ZZ8,ZZ9,ZZ10
      DATA NOU/52/
C
C     *** WRITE RANGE, DEPTH AND U(I)
          IF((RA2 .LT. PRTIN) .OR. (RA2 .GT. PRTCT)) GO TO 50
          WRITE(NCU,*)   RA2, ZR, U(IWZ)
   50 CONTINUE
C
C     *** DETERMINE NEXT RANGE AT WHICH TO WRITE SOLUTION
          XWR = XWR+WDR
C
      RETURN
      END
C
C
C
C
      SUBROUTINE ATTENU(U,ATT,IA,NA)
C
C     THIS SUBROUTINE APPLIES ARTIFICIAL ATTENUATION TO THE BOTTOM-
C     MOST NA GRID POINTS AS PER AESD PE MODEL BY BROCK  -  NORDA
C     TECH NOTE 12  -  JAN 78
C          (1)  ATTENUATION  MATRIX ATT IS CALCULATED IN SUBROUTINE
C               NEWMAT
C
      COMPLEX U(5000)
      DIMENSION ATT(5000)
C
      DO 10 I=1,NA
         U(IA+I) = U(IA+I) * ATT(I)
 10   CONTINUE
C
      RETURN
      END
C
C
C
C
      SUBROUTINE END (RA2)
C
C     THIS SUBROUTINE IS CALLED WHEN THE SOLUTION FIELD HAS REACHED
C     THE MAXIMUM RANGE SPECIFIED (RMAX).  THE SUBROUTINE SENDS
C     APPROPRIATE MESSAGES TO THE TERMINAL AND STOPS EXECUTION.
C
      WRITE(6,899) RA2
      WRITE(6,900)
      WRITE(6,901)
```

```
C
      STOP
 899  FORMAT(/,' THE PROGRAM HAS REACHED RANGE R =',F8.1,' METERS.')
 900  FORMAT ( ///,
     *   5X,' END OF RUN',/,
     *  10X,' (1)  OUTPUT PRINTER FILE:',/,
     *  15X,' (A)  A FILE WITH FORMATTED OUTPUT HAS BEEN GENERATED.',/,
     *  15X,' (B)  THIS FILE IS READY TO BE SENT TO THE PRINTER.',/,
     *  15X,' (C)  THE FILENAME AND FILETYPE FOR THIS FILE ARE:',/,
     *  15X,'          IFDOUT          PRINTER',//,
     *  10X,' (2)  OUTPUT PLOTTER FILE:',/,
     *  15X,' (A)  A FILE WITH UNFORMATTED OUTPUT HAS BEEN GENERATED.',/,
     *  15X,' (B)  THIS FILE CONTAINS INPUT DATA FOR THE PLOT ROUTINE.',/,
     *  15X,' (C)  THE FILENAME AND FILETYPE FOR THIS FILE ARE:',/,
     *  15X,'          IFDOUT          PLOTTER',/,
     *  15X,' (D)  THE FILENAME AND FILETYPE FOR THE PLOT ROUTINE ARE:',/,
     *  15X,'          PLOTIFD         FORTRAN ')
 901  FORMAT (
     *  15X,' (E)  TO PLOT THE DATA, GO TO A GRAPHICS TERMINAL AND',/,
     *  15X,'      ENTER COMMANDS AS FOLLOWS:',/,
     *  15X,'             DEF STOR 1M    ',/,
     *  15X,'             I CMS          ',/,
     *  15X,'             FORTGI PLOTIFD ',/,
     *  15X,'             PLOTIFD        ',/,
     *  15X,'      THEN FOLLOW PROGRAM PROMPTS.')
C
      RETURN
      END
```

LIST OF REFERENCES

1. Naval Underwater Systems Center Technical Report 4103, *Fast Field Program for the Multilayered Media*, by F.R. DiNapoli, 26 June 1971.

2. Naval Underwater Systems Center Technical Memorandum, *Signal Transmission in the Wavenumber Domain*, by R. Lauer, 5 June 1979.

3. Stamey, B.B., *Preliminary Investigation of the Environmental Sensitivity of Acoustic Signal Transmission in the Wavenumber Domain with Respect to Source Depth Determination*, Naval Postgraduate School, December 1982.

4. Naval Ocean Research and Development Activity Technical Note 12, *The AESD Parabolic Equation Model*, H.K. Brock, January, 1978.

5. Officer, C.B., *Introduction to the Theory of Sound Transmission with Application to the Ocean*, McGraw-Hill, 1958.

6. Buckner, H.P., "Use of calculated sound fields and matched-field detection to locate sound sources in shallow water," *J. Acoust. Soc. Am.*, v. 59, p. 368-373, February 1976.

7. Brekhovskikh, L.M., *Fundamentals of Ocean Acoustics*, Springer-Verlag, New York, 1982.

8. DeSanto, J.A., "Relation between the Solutions of the Helmholtz and Parabolic Equations for Sound Propagation," *J. Acoust. Soc. Am.*, v. 62, p. 295-297, August 1977.

9. DeSanto, J.A., *Topics in Current Physics: Ocean Acoustics*, Springer-Verlag, 1979.

10. Coppens, A.B., *An Introduction to the Parabolic Equation for Acoustic Propagation*, Naval Postgraduate School Technical Report, NPS61-83-002, November 1982.

11. Kinsler, L.E., Frey, A.R., Coppens, A.B., and Sanders, J.V., *Fundamentals of Acoustics Third Edition*, John Wiley and Sons, 1982.

12. Naval Underwater Systems Center Technical Report 6659, *IFD: An Implicit Finite-Difference Computer Model for Solving the Parabolic Equation*, by D. Lee and G. Botseas, 27 May 1982.

13. Jaeger, L.E., *A Computer Program for Solving the Parabolic Equation Using an Implicit Finite-Difference Solution Method Incorporating Exact Interface Conditions*, Naval Postgraduate School, September 1983.

BIBLIOGRAPHY

Bannister, R.W., "Low-frequency surface interference effects in long-range sound propagation," *J. Acoust. Soc. Am.*, v. 69, p. 76-83, January 1981.

Brigham, E.O., *The Fast Fourier Transform*, Prentice-Hall Inc.,1974.

Brock, H.K., Buchal, R.N., and Spofford, C.W., "Modifying the Sound Speed Profile to Improve the Accuracy of the Parabolic- Equation Technique", *J. Acoust. Soc. Am.*, v. 62, p. 543-552, September, 1977.

Clay, C.S. and H. Medwin, *Acoustic Oceanography: Principles and Applications*, John Wiley and Sons,1977.

DiNapoli, F.R. and R.L. Deavenport, "Theoretical and Numerical Green's function field solution in a plane multi-layered medium", *J. Acoust. Soc. Am.*, v. 67, p. 92-105, January 1980.

Hamming, R.W., *Digital Filters Second Edition*, Prentice-Hall Inc., 1977.

Naval Undersea Center Technical Report TP 488, *Low-Frequency Propagation Effects for Sources or Recievers Near the Ocean Surface*, by M.A. Pedersen, D.F. Gordon, and D. White, September 1975.

Ross Donald, *Mechanics of Underwater Noise*, Pergamon Press, 1976.

Urick, R.J., *Principles of Underwater Sound*, Second Edition, McGraw-Hill, 1975.

Urick, R.J., *Sound Propagation in the Sea*, Defense Advanced Research Projects Agency, 1979.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, CA 93943	2
3.	Chairman (Code 68Mr) Department of Oceanography Naval Postgraduate School Monterey, CA 93943	1
4.	Chairman (Code 63Rd) Department of Meteorology Naval Postgraduate School Monterey, CA 93943	1
5.	Prof. A.B. Coppens (Code 61Cz) Department of Physics Naval Postgraduate School Monterey, CA 93943	4
6.	Prof. C.R. Dunlap (Code 68Du) Department of Oceanography Naval Postgraduate School Monterey, CA 93943	3
7.	Lt. J.L. Blanchard 598-B Michelson Rd. Monterey, CA 93940	3
8.	Director Naval Oceanography Division Naval Observatory 34th and Massachusetts Avenue NW Washington, DC 20390	1
9.	Commander Naval Oceanography Command NSTL Station Bay St. Louis, MS 39522	1
10.	Commanding Officer Naval Oceanographic Office NSTL Station Bay St. Louis, MS 39522	1
11.	Commanding Officer Fleet Numerical Oceanography Center Monterey, CA 93940	1
12.	Commanding Officer Naval Ocean Research and Development Activity (Attn: Code 201) NSTL Station Bay St. Louis, MS 39522	1

13. Commanding Officer 1
Naval Environmental Prediction
Research Facility
Monterey, CA 93940

14. Chairman, Oceanography Department 1
U.S. Naval Academy
Annapolis, MD 21402

15. Chief of Naval Research 1
800 N. Quincy Street
Arlington, VA 22217

16. Office of Naval Research (Code 420) 1
Naval Ocean Research and Development
Activity
800 N. Quincy Street
Arlington, VA 22217

17. Scientific Liaison Office 1
Office of Naval Research
Scripps Institution of Oceanography
La Jolla, CA 92037

18. Library 1
CICESE
P.O.Box 4803
San Ysidro, CA 92073

19. Commander 1
Oceanographic Systems Pacific
Box 1390
Pearl Harbor, HI 96860

20. Commander (AIR-370) 1
Naval Air Systems Command
Washington, DC 20360

21. Commanding Officer 1
Attn: Mr R.B. Lauer, Code 323
Naval Ocean Research and Development
Activity
NSTL Station
Bay St. Louis, MS 39522

205370

Thesis
B54843
c. 1

Blanchard

A comparison of two acoustic parabolic equation transmission loss models for compatibility with the Wavenumber Technique in the determination of source depth.

205370

Thesis
B54843
c.1

Blanchard

A comparison of two acoustic parabolic equation transmission loss models for compatibility with the Wavenumber Technique in the determination of source depth.

AMERICAN BINDE
TOPEKA K

A comparison of two acoustic parabolic e
3 2768 002 13535 2
DUDLEY KNOX LIBRARY